Configurando um Squid "Ninja"

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Configurando um Squid "Ninja"

Introdução

A World Wide Web (WWW) é, sem a menor dúvida, a forma mais conhecida da internet. Tanto isso é verdade, que os leigos tem uma certa dificuldade em entender que a internet não se resume ao www. Sua popularidade e crescimento são explicados pela grande variedade de assuntos encontrados nela, pela facilidade de busca, simples entendimento, baixo custo e, via de regra, privacidade.

Em decorrência dessa grande procura alguns efeitos colaterais ocorrem. Não é incomum ouvir as pessoas dizerem que "a internet está lenta", ou os administradores de rede observarem seus backbones atingirem seus limites em horários de pico. Do lado dos servidores e ISPs (Internet Service Providers) também existe um lado que poucas pessoas pensam. Somente quem já passou pelo "Efeito Slashdot1" sabe do que estou falando. Em um momento de sobrecarga dos servidores, como ocorreu na última copa do mundo ou no tenebroso 11 de setembro, um sistema de caches bem planejado e distribuído seria muito bem visto pelos grandes portais.

A utilização de sistemas de cache, como o Squid, têm se mostrado excelentes para aliviar esses sintomas, reduzindo o tráfego na rede e, conseqüentemente, a latência da mesma.

Toda a idéia por trás de um sistema de caching é criar um grande banco de dados onde os sites mais populares ou acessados recentemente são armazenados para futuras consultas. Isso significa que se 10 usuários da sua rede tentarem acessar um mesmo site ao mesmo tempo, somente uma das conexões realmente irá ser feita a esse site. Todas as outras 9 vão se aproveitar do primeiro acesso e utilizar a página já em memória. Isso é um enorme ganho de desempenho para seu backbone local, para o backbone do ISP onde o site está armazenado e para o servidor que hospeda o mesmo.

Além disso, sua banda fica livre para que sites menos acessados, ou que não estejam no cache sejam baixados com maior velocidade.

Com um sistema de caching bem planejado e mantido, todos tem a ganhar.

Sobre o autor

Eri Ramos Bastos trabalha com Linux desde 1998, passando por diversas distribuições e fases diferentes do pinguim. Atualmente trabalha como consultor em soluções Linux / Unix e está disponível para ajudar a sua empresa a implantar Linux em todos os setores. Consultoria

Sobre esse documento

Esse documento é GPL, tendo sido baseado em diversas documentações disponíveis na web. Todas as marcas citadas aqui pertencem aos seus respectivos donos. A Página oficial desse documento é http://www.linuxman.pro.br/squid/ . Também está disponível em versão PDF.

Esse documento foi totalmente escrito em texto puro, utilizando VIM como Editor e posteriormente convertido em HTML por txt2tags e em PDF por htmldoc.

Changelog

11/11/2003

- Adicionada seção changelog

- Adicionada seção ToDo
- Modificacoes no Lay−out da pagina para tornar mais legível
- Corrigidos erros de formatação, adaptando melhor o documento ao txt2tags
- Criada versão PDF desse documento PDF

13/11/2003

- Acrescentado comando para entrar em modo Debug na seção utilidades públicas
- Inclusão de códigos de erro e saída do Squid na seção utilidades públicas
- Mais ajustes estéticos

05/02/2004

- Sintaxe mais limpa do grep para limpar o squid.conf

23/03/2004

- Acrescentada dica sobre rotate dos relatórios do Sarg

18/06/2004

- Impedindo ou Limitando o tamanho de uploads
- Atualização da URL do Sarg

ToDo

- Adaptar "Examinando o Squid.conf" para novos padrões txt2tags
- Corrigir problemas
- Adicionar autenticação com proxy transparente

O que esperar de um proxy/cache?

Podemos sumarizar os benefícios esperados em:

- Velocidade de acesso

  A melhor forma de verificar se o seu cache está sendo eficiente é pela velocidade. Um sistema de cache que não agrega velocidade não está cumprindo o seu papel.

- Disponibilidade

  De nada adianta um sistema veloz disponível apenas 2 horas por dia, ou mesmo que precise de um reboot a cada 2 semanas. Se o seu sistema de caching ou seu sistema operacional não tem uma alta disponibilidade, esse howto chegou em boa hora. Em casos de grandes instalações, ainda é preciso ir mais a fundo, buscando a altíssima disponibilidade. Redundância de servidores, backup, eliminação de ponto único de falha e disaster recover são uma exigência.

- Transparência ou Ostensividade

São conceitos específicos e que se adaptam a cada caso. Grandes instalações, ISPs e empresas não preocupadas com que seus usuários vêem ou fazem na internet devem preferir a transparência, onde o usuário desconhece ou não se sente afetado (exceto pelo ganho de velocidade) pela presença de um cache.

Por outro lado, empresas com uma política de segurança mais rígida, órgãos com informações críticas, ou mesmo pais querendo controlar o acesso de seus filhos a alguns sites, vão preferir a ostensividade.

- Capacidade de trabalhar com redes heterogêneas.

  Alguns sistemas de proxy/cache funcionam baseados com sistemas de autenticação especiais, feitos para rodar somente em uma plataforma, fazem integração com o serviço de diretórios daquele ou desse sistema ou exigem que o usuário esteja rodando a versão XYZ do fabricante ABC e deixam todos os outros a ver navios. Em uma instalação séria, é preciso que usuários de todas as plataformas que saibam como trabalhar com HTTP sejam bem atendidos.

  Isso é especialmente verdade quando não sabemos que tipo de plataforma irá utilizar nossa instalação.

- Simplicidade

  Deixando um pouco de lado o usuário e focando no administrador, é preciso ter consciência de que um sistema bom é um sistema fácil de administrar. O mais rápido, mais disponível e mais abrangente sistema de caching é totalmente inútil se somente uma pessoa no mundo souber lidar com ele.

E o Squid? Satisfaz todos esses pontos?

Em uma resposta rápida: Sim.

Veremos mais abaixo que todos os requisitos listados são atendidos com primazia pelo Squid.

O que é o Squid?

Squid é um proxy−cache de alta performance para clientes web, suportando protocolos FTP, gopher e HTTP.

O Squid mantém meta dados e especialmente objetos armazenados na RAM, cacheia buscas de DNS e implementa cache negativo de requests falhos.

Ele suporta SSL, listas de acesso complexas e logging completo. Por utilizar o Internet Cache Protocol, o Squid pode ser configurado para trabalhar de forma hierárquica ou mista para melhor aproveitamento da banda.

Podemos dizer que o Squid consiste em um programa principal − squid −, um sistema de busca e resolução de nomes − dnsserver − e alguns programas adicionais para reescrever requests, fazer autenticação e gerenciar ferramentas de clientes.

Podemos executar o Squid nas principais plataformas do mercado, como Linux, Unixes e Windows.

Porque utilizar um Proxy/Cache?

Podemos dizer que existem dois grandes motivos pelo qual se deve utilizar um PROXY/CACHE:

Controle de acesso

Com a internet cada vez mais acessível a pequenas e médias empresas, um número imenso de pessoas está se interligando a internet. Além de todos os benefícios trazidos por ela, como informação em tempo real, comunicação mundial a baixo custo, contato com possíveis clientes e fornecedores por todo o mundo, a mesma trouxe alguns problemas.

As pessoas tendem a passar cada vez mais tempo navegando por sites não relativos ao seu trabalho primário, acessam sites que não condizem com a política da empresa, utilizam a banda de internet destinada a serviços como WEB ou VPN e podem, em muitos casos, acabar infectando toda a rede da empresa com vírus e worms que são adquiridos em sites impróprios. Isso sem contar na ameaça sempre presente de propagação de downloads de softwares piratas e músicas, fatores que podem complicar a vida de uma empresa durante fiscalizações.

De acordo com a Rede Nacional de Ensino e Pesquisa (RNP) , 65% da largura de banda das empresas é utilizada em navegação WEB. E esse número tende a crescer.

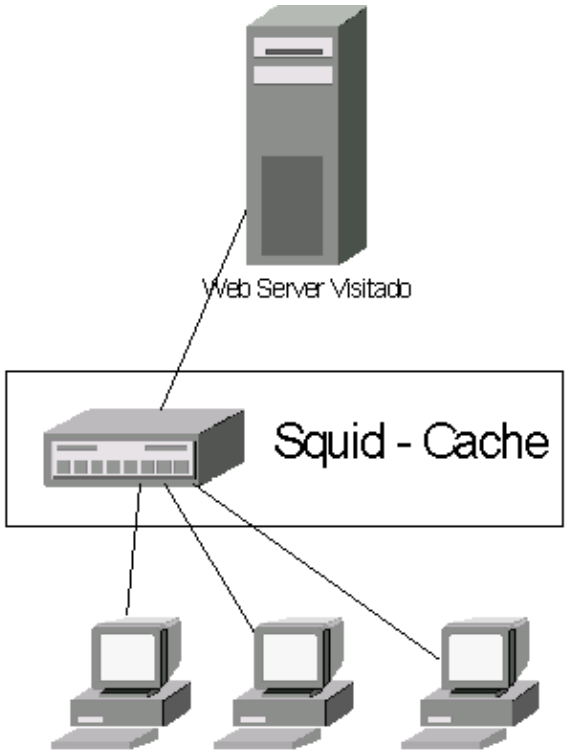
Performance

Como dissemos anteriormente, a internet está mais acessível para todos, fator causado pela ampla utilização das conexões de banda larga, como xDSL, Cable Modem, ISDN, etc.

Essas tecnologias são excelentes para pequenas e médias empresas, mas devido a suas características de velocidades diferentes de upstream e downstream (xDSL), compartilhamento de banda total (Cable Modem) ou baixo desempenho (ISDN), além da notável falta de qualidade das operadoras, tornam−se quase inúteis para grandes empresas e provedores de internet (ISPs).

Essas empresas são então levadas a utilizar sistemas de maior qualidade, como links por fibra ótica, satélites e rádio. Mas como se pode esperar, qualidade tem preço, e, nesse caso, bem salgado.

Visando aproveitar ao máximo essa banda de qualidade, a utilização de PROXY/CACHE torna−se quase que obrigatória. Ainda de acordo com a Rede Nacional de Ensino e Pesquisa (RNP) − 2, a utilização de PROXY/CACHE pode gerar uma economia entre trinta e cinqüenta por cento nos horários de pico. Isso significa que para um link de 2 Mbps que está operando a plena carga e considerando uma redução de 30 %, o mesmo produziria um ganho na banda agregada de aproximadamente 600 Kbps. Ou seja, a simples implementação de um PROXY/CACHE bem ajustado gera uma economia da ordem de milhares de Reais por mês para a empresa.



Conexões são feitas no Proxy, evitando saída à internet

Porque utilizar o SQUID?

O Squid está continuamente melhorando sua performance, além de adicionar novas features e ter uma excelente estabilidade em condições extremas.

Sua compatibilidade com várias plataformas e a imensa gama de software para analisar logs, gerar relatórios, melhorar o desempenho e adicionar segurança providos pela comunidade open source, combinados com ferramentas de administração simplificada e baseadas em web agregam grande valor ao produto.

Podemos ainda citar a capacidade de clustering, transparent proxy, cache de FTP e, é claro, seu baixo custo.

Para os mais corajosos, ou para os melhores programadores, não podemos deixar de dizer que o sistema é totalmente aberto, possibilitando a sua otimização no nível de código, além da otimização via configuração.

Protocolos utilizados − Rede e Aplicação.

O Squid busca por comunicação TCP (Transmission Control Protocol) e ICP (Internet Cache Protocol) em portas específicas. O TCP é usado para comunicação entre webservers e clientes, e o ICP para conversa entre servidores de cache. Para cada servidor (ou cliente), a configuração do Squid precisa fornecer uma única porta sobre a qual o Squid irá enviar as requisições (TCP ou ICP) e ouvir as respostas.

Como já dissemos anteriormente, o Squid trabalha apenas com FTP, gopher e http. Existe uma confusão muito comum entre pessoas que estão começando a trabalhar com o Squid em achar que poderão, através do Squid, configurar acesso a e−mails, ICQ, IRC, etc. Isso é totalmente equivocado, visto que não só é função do firewall trabalhar com o NAT (Network Address Translation), como também não faz sentido criar caches de

e−mails pessoais, mensagens do ICQ, etc.

Para configurar seu firewall apropriadamente para NAT, verifique a documentação de seu sistema operacional ou firewall.

Requisitos

A maior parte das configurações depende apenas do Squid. O proxy transparente também depende do sistema operacional e do firewall

A instalação padrão do squid, disponível na maior parte das distribuições, não consegue lidar com o controle de banda, sendo necessário recompilar o Squid.

Referências

Rede Nacional de Pesquisa (1) − http://www.rnp.br/newsgen/0103/wccp.shtml

Rede Nacional de Pesquisa (2) − http://www.rnp.br/arquivos/docgeral.html

Duane Wessels Home Page − http://www.life−gone−hazy.com/index−two.html

Firewall Linuxman (IPTABLES) − http://www.linuxman.pro.br/cgi−bin/firewall/

Survey of Web Caching Schemes for the Internet −

http://www.acm.org/sigcomm/ccr/archive/1999/oct99/Jia_Wang2.pdf

Instalando o Squid

O Squid pode ser instalado em uma imensa variedades de sistemas operacionais. Praticamente todos os Unixes com um bom compilador C/C++ pode gerar binários do Squid.

Sua popularidade, no entanto, nos poupa esse passo em muitas plataformas. Segue abaixo a forma de instalação nas mais populares plataformas do mercado.

Instalando via binário ou com facilidades do sistema

Se você não precisa de nenhuma feature muito sofisticada no seu squid (90% dos casos não precisa), não há porque instalar via código−fonte baixado do site do squid.

Vamos direto ao assunto:

Instalando em um sistema baseado em Red Hat Linux

Além de estar disponível nos CDs da distribuição, ainda é possível baixar as mais novas versões já empacotadas no sistema RPM (Red Hat Package Manager). Para isso acesse o link

http://www.rpmfind.net/linux/rpm2html/search.php?query=squid.ch=

E depois:

```
# rpm −ivh squid.x.y.z.rpm
```

Instalando em um sistema baseado em Debian

O Debian sempre prezou pela facilidade de instalação a atualização de pacotes, com seu sistema apt, que facilita muito a vida dos administradores. Para instalar o squid basta executar o comando:

```
# apt−get install squid
```

Instalando em um FreeBSD

Se você instalou o diretório de ports no FreeBSD, a instalação será simples, bastando utilizar os comandos abaixo:

```
# cd /usr/ports/www/squid25/
# make
# make all install
```

Ou por meio de um pacote pré−compilado:

```
# mount /cdrom #CD de instalação do FreeBSD
# mkdir /usr/ports/distfiles
# cp /cdrom/packages/All/squid−x.y.z.tgz /usr/ports/distfiles/ # onde  x.y.z é a versão.
# pkg_add −v /usr/ports/distfiles/squid−x.y.z.tgz
```

Instalando em um OpenBSD

Baixe o squid já compilado em http://www.openbsd.org/3.2_packages/i386/squid−2.5.PRE13.tgz−long.html

E depois:

```
# pkg_add squid.x.y.z.tgz
```

Instalando em um Windows 2000

Baixe o arquivo setup.exe do site do Cygwin (http://www.cygwin.com/setup.exe) e selecione o pacote do squid durante a instalação. Proceda como faria qualquer instalação em plataforma Microsoft.

Baixando o código−fonte

Caso queira o controle de banda, tópico avançado abordado aqui, instale o squid pelo fonte, de acordo com as instruções.

Na data de criação desse documento, a versão mais recente (estável) do squid era a 2.5STABLE1.

Verifique a versão mais recente em http://www.squid−cache.org/Versions/v2/.

```
# groupadd squid
# useradd −g squid −s /dev/null squid >/dev/null 2>&1
# wget http://www.squid−cache.org/Versions/v2/2.5/squid−2.5.STABLE1−src.tar.gz
# tar zxvf squid−2.5.STABLE1−src.tar.gz
# cd squid−2.5.STABLE1
# ./configure −−enable−delay−pools −−enable−cache−digests\
 −−enable−poll −−disable−ident−lookups −−enable−truncate \
 −−enable−removal−policies −−enable−arp−acl
# make all
# make install
# cd auth_modules/NCSA
# make
# make install
```

Limpando o squid.conf

O arquivo de configuração do squid é o squid.conf, normalmente ele se encontra em /etc/squid.conf ou em /usr/local/squid/etc/squid.conf. Caso não encontre o seu em nenhum desses lugares, procure−o com:

```
# locate squid.conf

   ou

# find squid.conf
```

Pode parecer fútil, mas uma limpeza inicial no arquivo squid.conf pode ser bem útil. O arquivo de configuração original tem, em média, 2000 linhas.

```
# cp squid.conf squid.conf.original
# egrep −v "^#|^$" squid.conf.original > squid.conf
```

Configurações básicas − ACLs

Como comentado mais tarde, toda a estrutura do Squid é baseada em listas de acessos. Vamos entrar em detalhes mais para frente. Por hora vamos criar uma lista de acesso básica para nossos usuários.

Vamos supor que nossa rede interna seja 192.168.5.0/24. Crie a seguinte linha no squid.conf, na seção de ACLs (TAG: acl):

```
acl rede_interna src 192.168.5.0/24
```

E a seguinte linha na seção de acesso (TAG: http_access)

```
http_access allow rede_interna
```

Referências

http://www.squid−cache.org/Doc/

http://www.serassio.it/SquidNT.htm

Transparent Proxy

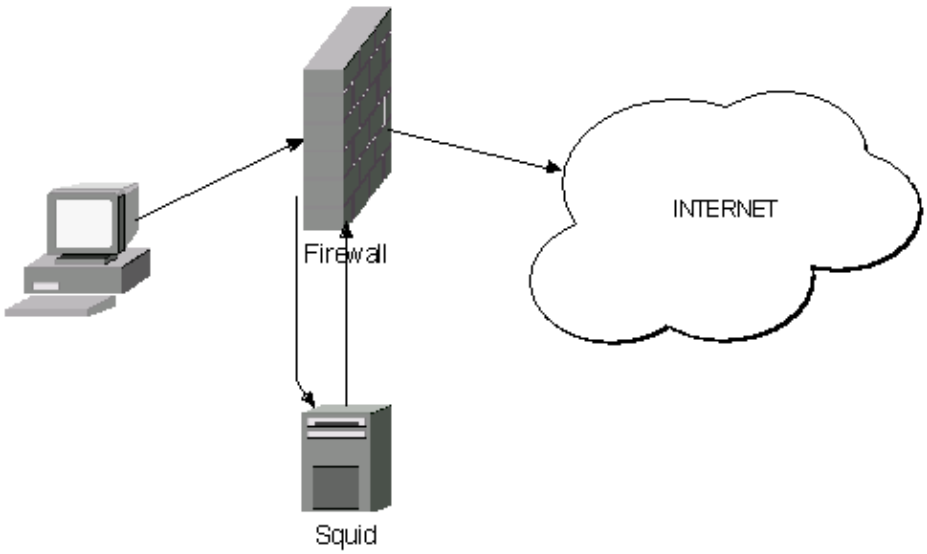
Esse recurso é muito útil para evitar que seus usuários "burlem" o proxy removendo as configurações do browser. Eles serão obrigados a passar pelo proxy, mesmo que as máquinas não estejam configuradas para tal. Extremamente recomendado, principalmente em casos de bloqueio de sites ou limitação de banda.

Experiências pessoais comprovam que usuários com um pouco mais de conhecimentos irão remover a configuração de proxy assim que o administrador sair da sala, seja por ignorância das funcionalidades, seja por medo de ser auditado ou simplesmente por má conduta.

Para ser possível o uso de proxy transparente com o Squid, o firewall deve ser configurado adequadamente. Se o seu firewall não está listado abaixo, procure na documentação do mesmo qual é a sintaxe equivalente.

Algumas pessoas desejam trabalhar ao mesmo tempo com autenticação e proxy transparente. Isso é possível de ser feito com uma interação entre o firewall e um cgi, ou algo do gênero.

Apesar de não ser do escopo do howto abranger regras de firewall específicas e programação, uma boa olhada no google e um pouco de pesquisa deve resolver o problema.



Como funciona o proxy transparente

Configurando o Squid

Vamos inserir as seguintes linhas:

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Configurando o iptables

Provavelmente você já tenha seu script de inicialização do firewall, sendo assim a única coisa necessária inserir essa linha nele:

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

Configurando o PF (OpenBSD)

Adicione a seguinte linha ao seu /etc/nat.conf (levando em consideração que sua interface interna seja fxp1)

```
rdr on fxp1 from any to any port 80 -> 127.0.0.1 port 3128
```

Configurando o IPFilter (FreeBSD)

Adicione as seguintes linhas ao seu /etc/rc.conf

```
ipfilter_enable="YES"
ipnat_enable="YES"
ipmon_enable="YES"
ipfs_enable="YES"
```

Adicione as seguintes linhas ao seu /etc/ipnat.rules (levando em consideração que o rl0 é sua interface interna)

```
rdr rl0 0/0 port 80 -> 127.0.0.1 port 3128 tcp
```

Referências

http://ldp.conectiva.com.br/HOWTO/mini/TransparentProxy.html

Bloqueando Sites indesejados

A partir de agora vamos começar a trabalhar com ACLs (Access Control Lists). O conceito de ACL é muito útil, por nos permitir trabalhar com níveis de acesso baseados em diversas informações.

Não é incomum que em uma instalação de Squid, a diretoria possa acessar qualquer site, a gerência não possa acessar determinados sites e os "peões" tenham acesso apenas ao site da empresa e de parceiros. Graças ao uso de ACLs e um pouco de imaginação e suor, podemos fazer todas essas restrições.

Todas as configurações de usuários, grupos, horários e SITES são configuradas em ACLs,

Vamos começar criando 2 ACLs que irão fazer o bloqueio dos sites indesejados. A ordem em que as ACLs aparecem é muito importante, por isso a ACL que bloqueia os sites deve ser a primeira a aparecer.

Procure no seu squid.conf onde começam a ser descritas as ACLs. Geralmente a primeira ACL a aparecer é:

```
acl all src 0.0.0.0/0.0.0.0
```

Criando os arquivos necessários

Vamos fazer o seguinte:

```
# mkdir /etc/squid/bloqueados
```

ou

```
# mkdir /usr/local/squid/etc/bloqueados
# touch /etc/squid/bloqueados/block.txt
```

ou

```
# touch /usr/local/squid/etc/bloqueados/block.txt
# touch /etc/squid/bloqueados/unblock.txt
```

ou

```
# touch /usr/local/squid/etc/bloqueados/unblock.txt
```

O arquivo block.txt irá conter todos os sites e palavras que você deseja bloquear e o unblock.txt todas as exceções. "Como assim?", você pergunta.

Vamos supor que você tenha bloqueado a palavra sexo. Então você não poderá entrar em www.sexo.com.br, mas também não poderá entrar em www.sexoesaude.com.br. Ora, mas esse segundo site é inofensivo, portanto não deveria ser bloqueado. Basta colocá−lo no unblock.txt.

Editando o squid.conf

Vamos ao squid.conf

Insira as linhas abaixo logo antes de acl all src 0.0.0.0/0.0.0.0:

```
  acl blockedsites url_regex -i "/etc/squid/bloqueados/block.txt"
  acl unblockedsites url_regex -i "/etc/squid/bloqueados/unblock.txt"
```

ou

```
  acl blockedsites url_regex -i "/usr/local/squid/etc/bloqueados/block.txt"
  acl unblockedsites url_regex -i "/usr/local/squid/etc/bloqueados/unblock.txt"
```

Agora procure no seu squid.conf a linha http_access deny all e coloque antes dela:

```
  http_access deny blockedsites !unblockedsites
```

DICA: O "!" Significa sempre negação de alguma coisa.

Referências

http://members.lycos.co.uk/njadmin/

http://web.onda.com.br/orso/

Bloqueio de Banners

Banner é uma coisa chata! Que me perdoem os anunciantes, mas eu não suporto banner nem pop−up. Mas o pior de tudo é que elas consomem banda e quase nunca ajudam o Squid, pois estão em constante mudança, impedindo o caching. Com a solução mostrada aqui, todos os banners serão substituídos por uma imagem pré−definida, podendo inclusive ser personalizada. Muito legal em empresas ou provedores de acesso em conjunto com o proxy transparente.

Baixando e instalando o Banner Filter

Baixe o Banner Filter do seu site oficial (http://phroggy.com/files/unix/bannerfilter−1.21.tar.gz).

```
# tar zxvf bannerfilter−1.21.tar.gz
# cd bannerfilter−1.21
```

Mova o conteúdo do diretório www para algum lugar acessível em seu web server. Esses arquivos PRECISAM estar acessíveis ao squid via HTTP. É importante ressaltar que o sistema perde o sentido se o servidor http não for a mesma máquina que o Squid está.

Mova todo o resto para /etc/squid/bannerfilter ou /usr/local/squid/etc/bannerfilter

Edite o redirector.pl. Se você não tem o perl no local padrão (/usr/bin/perl), mude a primeira linha (ou crie um symlink).

Mova o bannerfilter.conf para o /etc

Mude as variáveis $DATA e $WWW como indicado nos comentários.

Opcionalmente, mude também $LOG e $BANNERGIF como indicado.

Teste o redirector.pl digitando alguma coisa e veja se recebe essa coisa de volta.. Pressione Crtl−C para parar. Não pule esse passo, pois nele você poderá descobrir erros.

Rode o script update.sh para atualizar as listas de banners. É interessante fazer isso constantemente.

Editando o squid.conf

Procure pela seção que fala sobre redirect (TAG: redirect_program) e insira a linha:

```
redirect_program /etc/squid/bannerfilter/redirector.pl
```

ou

```
redirect_program /usr/local/etc/bannerfilter/redirector.pl
```

DICA: É possível também editar as imagens, de forma a torna−las personalizadas para sua empresa.

Os banners irão aparecer dessa forma após tudo instalado

Referências

http://phroggy.com/bannerfilter/

Protegendo usuários com antivírus

Essa solução deve ser usada apenas em pequenas instalações, como um adicional de segurança. Se a sua rede local tem antivírus nas estações e no servidor de domínio, arquivos, etc, eu não recomendo essa feature. Além de exigir muito da máquina, ainda não está totalmente estável. A solução proposta aqui é utilizar o Viralator.

Pré−requisitos

Será necessário que o Squid redirecione determinados downloads e URLs para o Viralator, de forma que precisamos do Squirm Instalado na máquina. Baixe−o em http://squirm.foote.com.au/squirm−1.0betaB.tar.gz

```
# tar zxvf squirm−1.0betaB.tar.gz
# cd squirm−1.0betaB
# cd regex
# ./configure
# make clean
# make
# cp −p regex.o regex.h
```

Anote o resultado desse comando:

```
# id `grep cache_effective_user /etc/squid.conf |cut −d " " −f3`
# cd ..
```

Edite o arquivo Makefile e substitua as aparições adequadas de "root" pelo usuário e grupo anotados acima.

```
# make
# make install
```

Além do Squirm, é necessário que o Apache e o apache−suexec estejam instalados. Procure uma documentação sobre o Apache para maiores detalhes.

E, como não poderia deixar de ser, um antivírus faz−se necessário. Atualmente o Viralator tem suporte à:

AntiVir

AVP

RAV

Inoculate

Sophos Sweep

McAfee

Trend

Viralator

Após ter o squirm instalado, adicione as seguintes linhas no seu arquivo squirm.paterns:

```
abortregexi (^http://[192.168.0.1].*)
abortregexi (^http://[cache1.empresa.com.br].*)
regexi (^.*\.zip$) http://[192.168.0.1]/cgi-bin/viralator.cgi?url=|\1
regexi (^.*\.doc$) http://[192.168.0.1]/cgi-bin/viralator.cgi?url=|\1
regexi (^.*\.exe$) http://[192.168.0.1]/cgi-bin/viralator.cgi?url=|\1
```

Onde: 192.168.0.1 é o IP do seu proxy e cache1.empresa.com.br é o FQDN2 do mesmo.

Repita a linha que faz referência a extensão para todos os tipos de arquivos que quiser escanear por vírus.

Edite agora o arquivo squid.conf adicionando um redirecionamento:

```
redirect_program /usr/squid/bin/squirm
redirect_children 10
```

Crie um usuário e grupo para uso do suexec e adicione−os ao seu arquivo de configuração do apache (normalmente httpd.conf)

```
< VirtualHost 192.168.0.1>
ServerAdmin webmaster@empresa.com.br
DocumentRoot /var/www/
ServerName cache1.empresa.com.br
ErrorLog logs/error_log
TransferLog logs/access_log
ScriptAlias /cgi-bin/ /usr/local/viralator/cgi-bin/
User viralator
Group viralator
</VirtualHost>
```

Onde: /usr/local/viralator/cgi−bin/ deve ser o seu diretório de cgis e viralator o nome do usuário e grupo que você criou.

Crie um diretório chamado downloads acessível ao apache − Algo como /var/www/downloads no Debian − mude suas permissões para 755.

Baixe o Viralator em http://viralator.loddington.com/downloads/viralator−09pre2.zip e execute os comandos:

```
# unzip viralator−09pre2.zip
# cp viralator−09pre2.cgi /usr/local/viralator/cgi−bin/viralator.cgi
# chown viralator.viralator −R /usr/local/viralator/cgi−bin/
# chmod 755 /usr/local/viralator/cgi−bin/viralator.cgi
```

Edite o arquivo /usr/local/viralator/cgi−bin/viralator.cgi e verifique se todos os caminhos de programas estão corretos.

Referências

http://viralator.loddington.com/

http://squirm.foote.com.au/

Autenticando usuários

É um recurso bem interessante para controle pessoal de usuários. Isso permite que você crie ACLs individuais e gere LOGs de qualidade bem superior.

Existem diversos métodos de autenticação, sendo interessante averiguar exatamente o que você irá precisar. Na maioria dos casos, o ncsa_auth resolve o problema.

ncsa_auth

O ncsa_auth é a alternativa mais simples. Ele está disponível junto com o squid e pode ser implementado rapidamente. É a solução ideal para pequenas e média instalações e redes com arquitetura de grupo de trabalho.

Editando o squid.conf

Procure pela seção que fala sobre autenticação (TAG: authenticate_program) e insira as linhas:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/
auth_param basic children 5
auth_param basic realm Digite seu Login
```

Criando um arquivo de senhas

O arquivo /etc/squid/passwd não existe por padrão. Para cria−lo vamos fazer:

```
# touch /etc/squid/passwd
```

Adicionando usuários

Para adicionar novos usuários basta fazer:

```
# htpasswd /etc/squid/passwd USUARIO
```

e confirmar a senha duas vezes.

Nota: Dependendo da sua distribuição, o ncsa_auth pode estar em vários lugares, como /usr/bin, /usr/sbin e assim por diante! Verifique onde está a sua e coloque as linhas acima de acordo!

Quanto ao authenticate_children 5, é o suficiente se sua rede não é muito grande. Mude o valor de acordo com suas necessidades.

Agora vamos editar novamente a ACL de nossa rede interna (aquela da seção 2.3)

```
acl rede_interna src 192.168.5.0/24
acl rede_interna proxy_auth REQUIRED
```

smb_auth

O smb_auth é uma ótima opção para quem tem uma rede um pouco maior ou trabalha com ambientes Windows Client/Server. Devido a sua integração com o PDC, facilita muito a vida do administrador.

Nota: É necessário que o samba esteja instalado na máquina do Squid para utilizar essa opção. Ele não precisa estar configurado ou ativado.

Instalando o smb_auth

Baixe o smb_auth em http://www.hacom.nl/~richard/software/smb_auth−0.05.tar.gz

```
# tar zxvf smb_auth−0.05.tar.gz
# cd smb_auth−0.05
```

Edite o arquivo Makefile e tenha certeza de que os parâmetros SAMBAPREFIX e INSTALLBIN estão corretos.

```
# make
# make install
```

Configurando o PDC

Para controlar o acesso por usuários e grupos, o smb_auth lê o arquivo \netlogon\proxyauth em um dos controladores de domínio previamente informado. Se a leitura desse arquivo retorna um "allow", então o acesso é liberado. Caso contrário, negado.

Crie um arquivo chamado proxyauth no compartilhamento NETLOGON de seu PDC (dê preferência ao primário). Esse arquivo deve conter unicamente a palavra "allow" (sem as aspas) e dê permissão de leitura para os grupos e usuários que deseja permitir acesso.

Configurando squid.conf

Adicione as seguintes linhas:

```
auth_param basic program /usr/local/bin/smb_auth −W DOMINIO −U 192.168.5.24
auth_param basic children 5
auth_param basic realm Digite seu Login
```

Onde: DOMINIO é o domínio do PDC e 192.168.5.24 é o IP do mesmo.

Referências

http://squid.visolve.com/squid24s1/externals.htm#authenticate_program

http://web.onda.com.br/orso/ncsaplus.html

http://www.hacom.nl/%7Erichard/software/smb_auth.html

http://www.linux.trix.net/dicas_squid_nt.htm

Controle de Banda

Esse é um feature muito útil para quem tem uma banda estreita, ou simplesmente tem prioridades para sua banda.

O recurso do squid que usamos aqui é chamado de delay pools. É necessário que o squid tenha sido compilado com essa opção ativa, conforme instruções da seção 2.

Editando o squid.conf

Vamos adicionar algumas linhas. A primeira vai evitar que haja restrição de banda internamente, por isso não deixe de colocá−la.

```
acl controle1 url_regex −i 192.168.5
acl controle2 url_regex −i ftp .exe .mp3 .tar.gz .gz .zip .rar .avi .mpeg .mpg .qt .ram .rm .is
delay_pools 2
delay_class 1 2
delay_parameters 1 −1/−1 −1/−1
delay_access 1 allow controle1
delay_class 2 2
delay_access 2 allow rede_interna
delay_access 2 allow controle2
```

Referências

http://www.tldp.org/HOWTO/Bandwidth−Limiting−HOWTO/index.html

http://www.linuxit.com.br/modules.php?name=Sectionsrtid=232

Brincando com ACLs

Vamos tentar agora explorar mais a fundo as possibilidades que as ACLs nos fornecem. É bom lembrar que várias ACLs podem ser combinadas, sendo isso um grande gerador de problemas. Faça suas regras com muita atenção.

Utilizando IPs e redes

Isso é o arroz−com−feijão das ACLs. Limitar por IP e/ou rede. Vamos por exemplos para simplificar:

```
acl ip_do_diretor src 192.168.5.5
acl ips_da_diretoria src 192.168.5.5 192.168.5.6 192.168.5.7 168.5.8
acl rede_do_rh src 192.168.6.0/24
acl rede_do_cpd src 192.168.7.0/255.255.255.0
```

Usando ACLs externas

O recurso de ACL externa é muito útil para um tratamento melhorado de algum recurso que não é compreendido por ACLs normais.

Uma ACL externa pode ser escrita em qualquer linguagem. Ela deve sempre retornar OK para o stdout caso a condição seja satisfeita, ou ERR também para o stdout caso ela não seja satisfeita.

Vou mostrar aqui um exemplo onde a diretoria deve acessar qualquer coisa, mas os usuarios normais sao submetidos a certas restrições. Levo em consideração que o usuário já está autenticado.

```
external_acl_type checa_diretoria %LOGIN /etc/squid/modulos/diretoria.sh
acl diretoria external checa_diretoria
```

Arquivo /etc/squid/modulos/diretoria.sh (deve ser executável)

```
#!/bin/bash
while read linha
do
        if [ `grep −i $linha /etc/squid/users/diretoria` ]
        then
                        echo OK
        else
                        echo ERR
        fi
done
```

Esse script verifica se o usuário autenticado pertence à diretoria. Para que um usuário seja reconhecido como diretoria, seu username deve estar dentro do arquivo /etc/squid/users/diretoria .

Trabalhando com domínios

Esse tipo de ACL tem que ser utilizada com cuidado. Tentar bloquear o acesso a chat em portais com essa opção também pode acarretar em acesso negado a sites de notícias ou de interesse geral. Todos os sub−domínios e hosts abaixo do domínio principal são afetados pela ACL.

```
acl GEOCITIES dstdomain geocities.com
```

Restringindo por horário

```
acl expediente time MTWHF 9:00-18:00
acl final_de_semana time SA 8:00-13:00
```

Onde:

Sigla	Dia
S	Domingo
M	segunda−feira
T	terça−feira
W	quarta−feira
H	quinta−feira
F	sexta−feira
A	sábado

Expressão regular na URL

Aqui podemos fazer milhares de coisas, desde que conheçamos muito bem expressões regulares. Para saber mais sobre elas, procure o livro "Expressões Regulares − Guia de Consulta Rápida" ou pesquise na internet.

```
acl jogos url_regex jogos
```

MAC Address

Para utilizar essa opção, o Squid deve ser compilado com os parâmetros "−−enable−arp−acl", como feito em nossa instalação via source.

```
acl administrador arp XX:XX:XX:XX:XX:XX
```

Onde: XX:XX:XX:XX:XX:XX é o MAC Address da placa de rede do administrador.

Limitando o número de conexões por usuário

Se quiser limitar o número de sessões que cada usuário abre de uma única vez, podemos utilizar o recursos de máximo de conexões.

```
acl CONEXOES maxconn 10
http_access deny CONEXOES rede_interna
```

Impedindo ou Limitando o tamanho de uploads

Diversas empresas tem a necessidade de impedir que seus usuários dêem upload de arquivos para webmails, discos virtuais ou algum outro tipo de repositório na internet. O grande problema é que o método utilizado para fazer estes uploads é o POST, também utilizado para preenchimento de formulários, pesquisas, logins e senhas, etc. Isso impede o bloqueio total do método. Como fazer então?

A opção mais lógica seria limitar o tamanho de um POST para um número suficientemente grande para permitir seu funcionamento normal e suficientemente pequeno para impedir o upload de arquivos. Para isso usamos a tag **request_body_max_size**, abordada um pouco mais abaixo.

No entanto essa tag tem uma falha, por não ser compatível com ACLs, ela limitará todos os usuários no que for determinado, situação normalmente incômoda.

Segue um script que encontrei na internet (referência abaixo), que nos permite criar ACLs baseadas nesse parâmetro. Vamos chamá−lo de /etc/squid/modulos/size.sh

```
#!/bin/sh
while read line; do
  set −− $line
  length="$1"
  limit="$2"
  if [ "$length" −le "$2" ]; then
      echo OK
   else
      echo ERR
   fi
done
```

Depois basta criar uma ACL assim:

```
external_acl_type request_body %{Content−Length} /etc/squid/modulos/size.sh
acl request_max_10KB request_body 10240
```

Com isso limitamos o tamanho do upload para 10KB, o que deve ser suficiente para preenchimento de um formulário, mas pouco para um upload.

Referências

http://squid.visolve.com/squid24s1/access_controls.htm

http://www.secforum.com.br/article.php?sid=1259

http://www.mail−archive.com/squid−users@squid−cache.org/msg16568.html

Criando um arquivo de configuração automática

Para facilitar a vida dos usuários (e do administrator), podemos criar um arquivo de configuração automática que será colocado nos browsers dos clientes. Dessa forma todos terão seu proxy reconfigurado dinamicamente em caso de mudanças, sem a necessidade de intervenção em cada máquina.

Esse arquivo deve ser acessível via web e, via de regra, chama−se proxy.pac .

Vamo supor que seu proxy esteja rodando no servidor 192.168.5.1 na porta 3128 e você não deseje que ele seja utilizado nas páginas do seu domínio (empresa.com.br):

```
function FindProxyForURL(url, host)
    {
            if (isPlainHostName(host) ||
                dnsDomainIs(host, ".empresa.com.br"))
              return "DIRECT";
           else
                return "PROXY 192.168.5.1:3128; DIRECT";
   }
```

Referências

http://wp.netscape.com/eng/mozilla/2.0/relnotes/demo/proxy−live.html

Gerando relatórios

Muitas empresas e instituições exigem dos administradores relatórios do uso da internet. Isso pode ser facilmente conseguido com algumas ferramentas.

SARG

Desenvolvido pelo brasileiro Pedro Orso, ele transforma o log do squid em um relatório html legível e completo.

Instalação

```
# wget http://web.onda.com.br/orso/sarg-1.4.tar.gz
# tar zxvf sarg-1.4.tar.gz
# cd sarg-1.4/
# ./configure
# make
# make install
```

Configuração

Por padrão o sarg é instalado em /usr/local/sarg. Nesse diretório encontramos o arquivo sarg.conf entre as muitas opções, recomendo as seguintes:

- language Portuguese
- access_log /var/log/squid/access.log
- title "Relatório de uso da internet"
- temporary_dir /tmp
- output_dir /var/www/squid-reports
- resolve_ip no
- user_ip yes
- topuser_sort_field BYTES reverse
- topsites_num 100
- max_elapsed 28800000

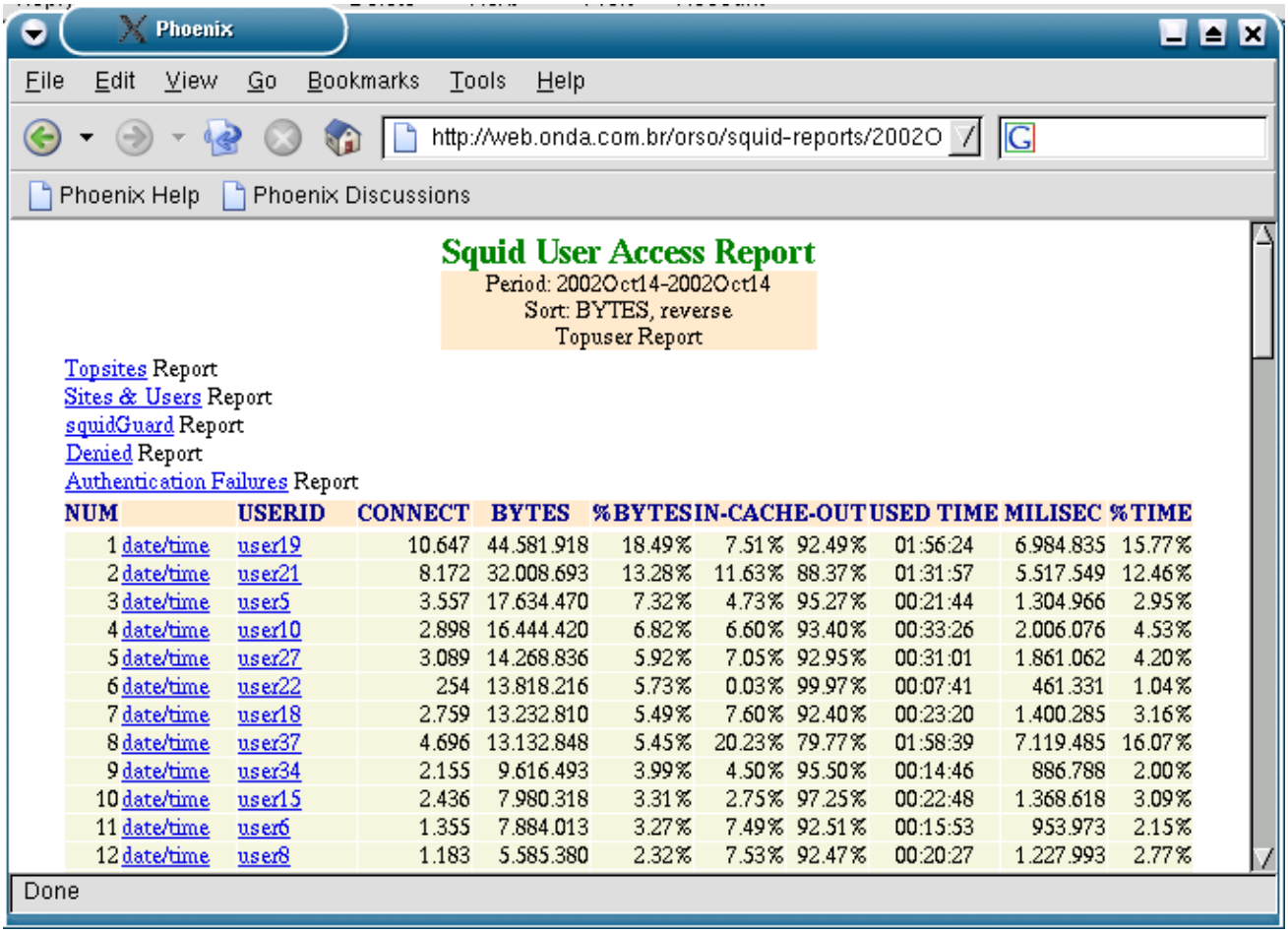
Sendo importante destacar:

Comandos	Descrição
access_log	Indica o arquivo de log do squid
output_dir	Indica onde será gerado o html. É recomendável que seja em um local acessível pelo seu http server
resolve_ip	Evita que o sarg tente fazer resolução de DNS
user_ip	Se você não estiver utilizando autenticação por usuário, coloque "no" . Se estiver, coloque "yes"
topsites_num	Quantidade de sites que você quer ver como os TOP de acessos

Gerando os relatórios

Depois de configurar o sarg.conf, basta gerar os relatórios com o comando

```
# sarg
```



Exemplo de relatório do SARG

Dica

Os relatórios do Sarg ocupam um imenso espaço em disco, principalmente pelo falto de não ter um rotate nem comprimir os HTMLS. Podemos contornar isso colocando em nossa crontable:

```
find /var/www/squid-reports/ -name "*.html" -type f -mtime +30 -exec bzip2 {} \;
find /var/www/squid-reports/ -name "*.bzip2" -type f -mtime +180 -exec rm -rf {} \;
```

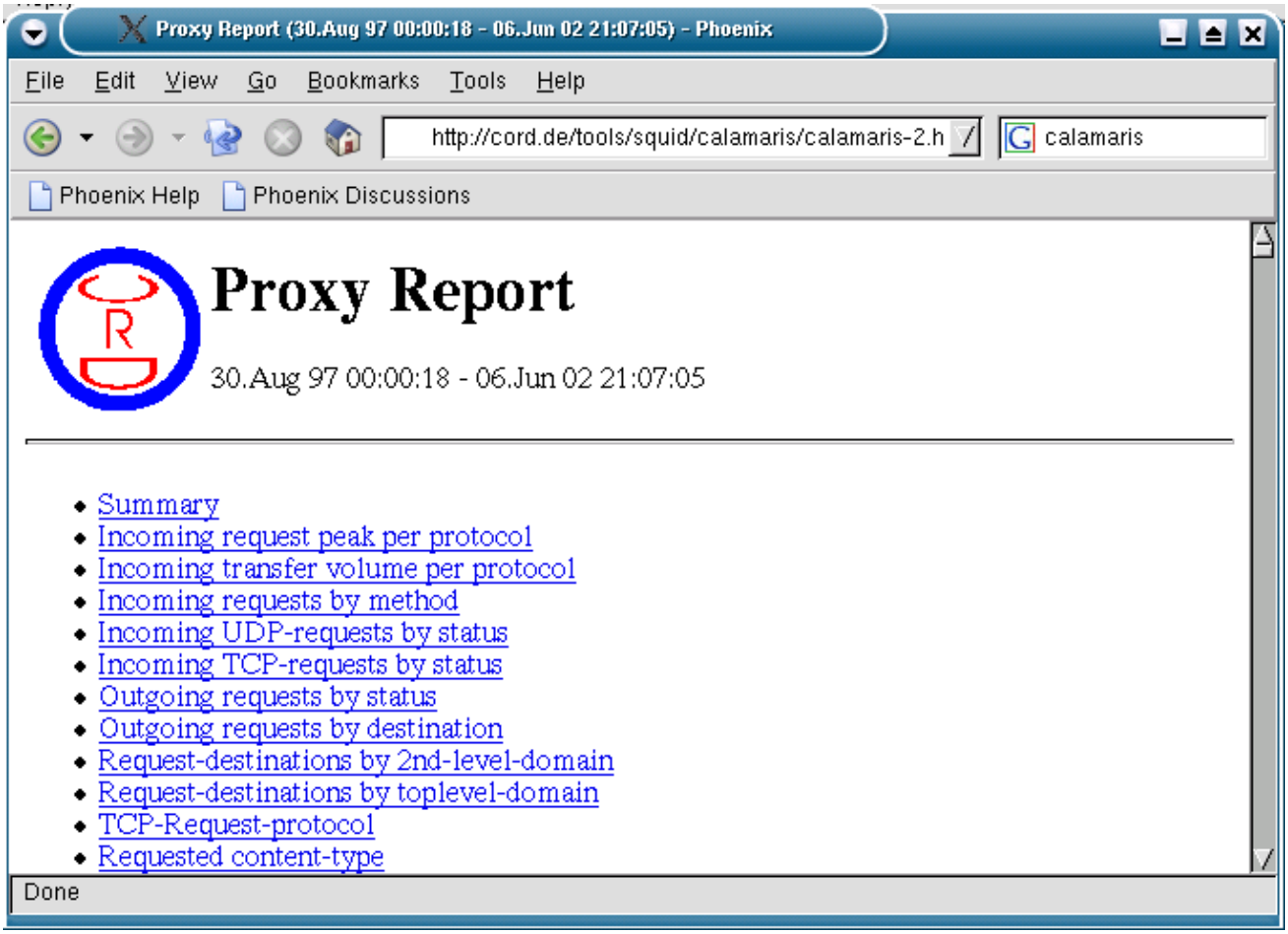
Calamaris

O Calamaris é um tradicional programa de análise de log e geração de reports para o squid. Seu funcionamento é simples e não exige instalação. Apenas é necessário ter o perl instalado na máquina.

Baixando e rodando

Direto e reto:

```
# wget http://cord.de/tools/squid/calamaris/calamaris-2.57.tar.gz
# tar zxvf calamaris-2.57.tar.gz
# cp calamaris /usr/bin
# cat /var/log/squid/access.log | calamaris -F html
```



Exemplo de relatório do Calamaris

Squid Graph

Ao estilo MRTG, esse analisador é ideal para uso em grandes caches, onde o importante não é saber quais usuários acessaram que site, quem teve acesso negado e etc. O objetivo aqui é analisar volume de tráfego e eficiência em grande escala.

Instalação

Esse programa exige a presença do módulo perl GD (http://stein.cshl.org/WWW/software/GD/). Instale-o antes de começar os passos abaixo.

Satisfeitas as dependências, baixe o Squid Graph de http://squid-graph.securlogic.com/files/stable/squid-graph-3.1.tar.gz e faça a "operação padrão":

```
# wget http://squid-graph.securlogic.com/files/stable/squid-graph-3.1.tar.gz
# tar zxvf squid-graph-3.1.tar.gz
```

e depois:

```
# mv squid-graph-3.1 /usr/local/squid-graph
# chmod +x /usr/local/squid-graph/bin/*
```

Como é um sistema feito em perl, não é necessário compilar.

Criando os gráficos

Para gerar um gráfico padrão:

```
# /usr/local/squid-graph/bin/squid-graph --output-dir=/destino/ \
< /var/log/squid/access.log
```

Para gerar um gráfico acumulativo:

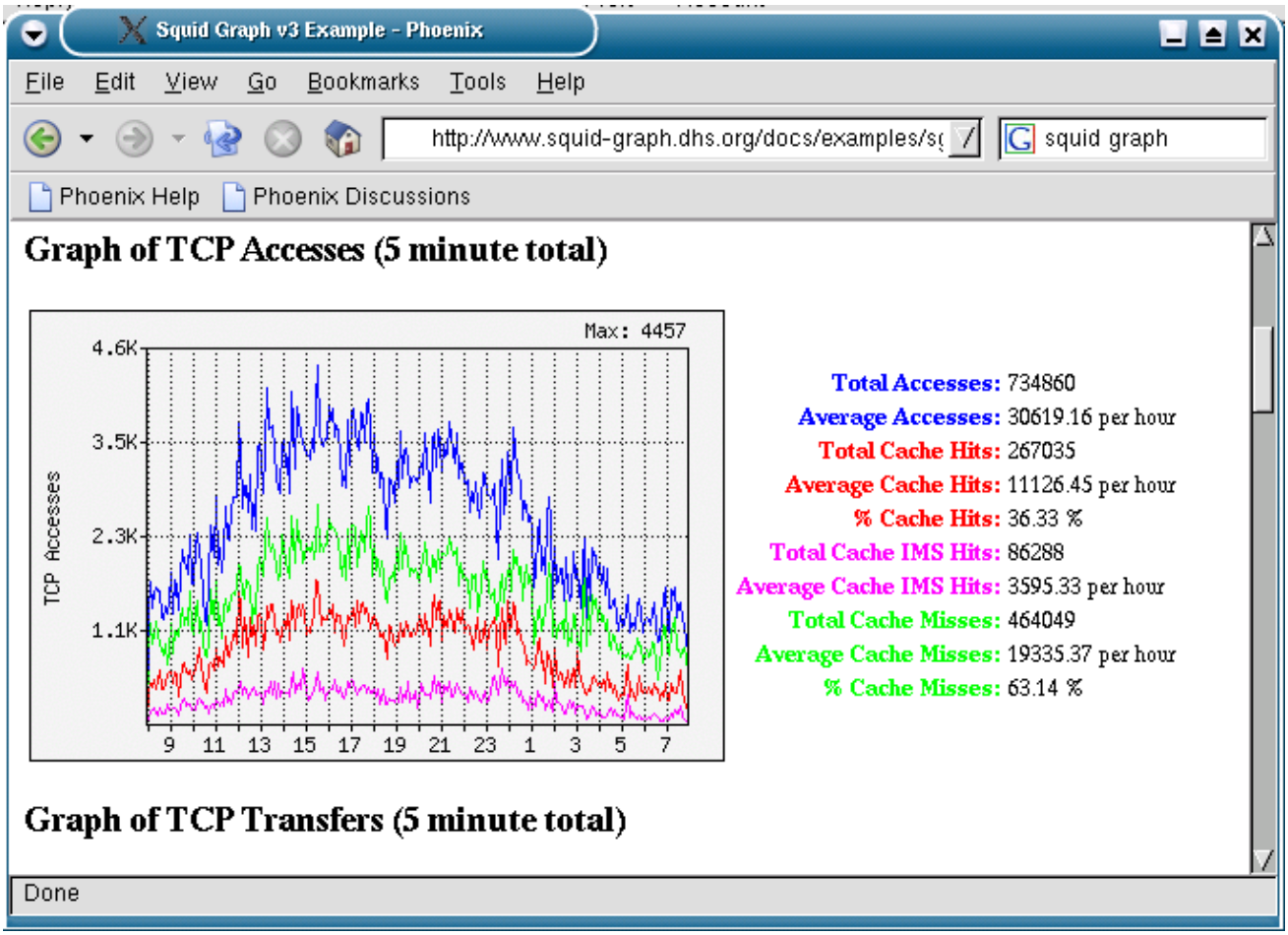
```
# /usr/local/squid-graph/bin/squid-graph --cumulative \
--output-dir=/destino/ < /var/log/squid/access.log
```

Para gerar um gráfico somente de TCP:

```
# /usr/local/squid-graph/bin/squid-graph --tcp-only \
--output-dir=/destino/ < /var/log/squid/access.log
```

Para gerar um gráfico somente de UDP:

```
# /usr/local/squid-graph/bin/squid-graph --udp-only \
--output-dir=/destino/  /var/log/squid/access.log
```



Exemplo de relatório do SquidGraph

Referências

Sarg's Home Page

Trabalhando com Hierarquias

Cache hierárquico é a extensão lógica do conceito de caching. Um grupo de caches podem se beneficiar do compartilhamento de seus dados entre si sobremaneira. Isso é facilmente explicável quando pensamos em termos regionais.

Exemplo: Sua empresa está estabelecida em um prédio junto com diversas outras. Esse prédio é atendido pelas empresas de telecom A, B e C.

Nesse caso, quando um usuário da empresa 1 deseja acessar um site, ele vai até seu proxy, que busca o site e o armazena, agilizando a consulta de todos os outros usuários dessa mesma empresa. Isso acontece também na empresa 2, 3 e etc.

Fica fácil de visualizar que se todas as empresas interligassem localmente seus proxies, todas teriam ganho.

Na realidade, essa sinergia entre pequenas empresas não existe. Mas quando falamos de grandes empresas e grandes backbones, cada 1 MB economizado com caching é 1 MB ganho em outros serviços.

Além de trabalhar com o conceito de árvore, onde existe um cache principal e outros ligados a ele, o Squid trabalha também com um conceito parecido com grupo de trabalho, onde todos os servidores se consultam mutuamente.

Toda a comunicação entre os caches é feita via ICP

Entendendo o ICP

O ICP foi desenvolvido como parte fundamental do projeto Harvest (Pai do Squid). Seu objetivo é prover um método rápido e eficiente de obter−se comunicação entre servidores cache.

O ICP permite que um cache pergunte a outro se ele tem uma cópia válida de um determinado objeto, aumentando a possibilidade de encontrar aquele objeto já cacheado. Adicionalmente, o ICP permite que requisições trafeguem entre servidores filhos em uma estrutura de árvore.

Além do controle de cache, o ICP também gera indicações do estado da rede. O não recebimento de uma resposta ICP normalmente indica que a rota está congestionada ou que o outro host está morto. Além disso, a ordem de chegada de uma resposta ICP pode indicar quais hosts estão com uma distância lógica menor ou com menos carga.

As mensagens ICP são geralmente bem pequenas, com cerca de 66 bytes. Em uma estrutura hierárquica, normalmente tem−se mais trocas de mensagens ICP do que HTTP.

Fazendo roteamento por domínios

Essa feature, apesar de simples, pode melhorar muito o desempenho de grandes instalações.

Vamos imaginar um caso em que existam 1 cache principal ligado a 3 outros caches. Vamos dizer também que temos uma imensa massa de usuários fazendo requisições a 3 grandes portais e ao mundo em geral.

A configuração seria algo assim:

```
cache_host_domain cache1 portalxpto.com
cache_host_domain cache2 portalxing.com
cache_host_domain cache3 portalling.com
cache_host_domain cache4 !portalxing.com ! portalxpto.com !portalling.com
```

Sendo que o cache4 será o responsável por todos os domínios que não sejam os 3 anteriores.

Roteando por protocolo

Podemos também definir qual será a rota tomada baseando−se em protocolo.

```
acl FTP proto FTP
acl HTTP proto http
cache_host_acl cache1 FTP
cache_host_acl cache2 HTTP
```

Pai e filho

O Caso exista um único servidor pai e diversos filhos, a configuração será:

```
    cache_host cache1 parent 3128 3130 default
```

Pais e filho

Em uma situação ideal, existem diversos servidores. A escolha sobre qual utilizar será baseada no método round robin.

```
cache_host cache1 parent 3128 3130 round−robin no−query
cache_host cache2 parent 3128 3130 round−robin no−query
cache_host cache3 parent 3128 3130 round−robin no−query
```

Referências

http://www.squid−cache.org/Doc/Hierarchy−Tutorial/tutorial.html

Utilizando o Squid como proxy reverso

Uma feature muito útil, mas por vezes pouco explorada do Squid é sua capacidade de trabalhar com proxy reverso. Isso signifca que, além de armazenar objetos remotos, criando toda uma série de vantagens já discutidas aqui, ele também pode armazenar objetos de um servidor web interno, aliviando seu uso e provendo maior segurança. Aqui o Squid literalmente trabalha como se fosse um servidor web.

Essa feature se mostra muito útil quando temos um web server com load alto, exigindo a ampliação da máquina ou criação de um cluster. Também é útil quando o servidor web utilizado pela empresa é conhecidamente inseguro e se mostra como um ponto fraco na empresa. O mesmo será "protegido" em alguns aspectos pelos Squid.

No momento de desenvolvimento da página já deve−se planejar uma futura implementação de Squid, cuidando para nunca desenvolver conteúdo unfriendly para o web caching. Tanto conteúdo estático quando dinâmico pode ser utilizado. O conteúdo dinâmico não será armazenado, enquanto o estático e coisas como imagens ficarão no Squid, aliviando o tráfego no web server para o conteúdo dinâmico ter maior fluidez.

Configuração de proxy reverso

A configuração é simples. Siga os passos abaixo:

```
http_port 80
httpd_accel_host 192.168.0.51
httpd_accel_port 80
httpd_accel_single_host on
httpd_accel_uses_host_header off
```

Onde:

Parametro	Objetivo
http_port 80	Número da porta onde o Squid irá escutar
httpd_accel_host 192.168.0.51	IP do servidor Web interno
httpd_accel_port 80	Porta onde o web server está escutando
httpd_accel_single_host on	Ativa o squid para somente um web server atrás
httpd_accel_uses_host_header off	É importante manter essa opção OFF, visto que ela altera os headers

Referências

http://squid.visolve.com/white_papers/reverseproxy.htm

Otimizando o Squid

Vamos listar algumas dicas para tornar o desempenho de seu Squid. Algumas delas são genéricas, como aumentar a memória alocada pelo Squid, outras são específicas, como utilizar um determinado sistema de arquivos no Linux.

Especificando o Hardware

Essa etapa é importante no início do projeto. O ideal é traçar um perfil de como é e como será em 1 ano o volume de uso desse hardware.

Procure sempre utilizar hardware que permita crescimento, especialmente em memória e armazenamento. Evite instalar servidores já com todos os bancos de memória usados ou no máximo.

Pequenas instalações dispensam HD (disco) SCSI, uma opção que já fica inviável em instalações maiores.

Ao utilizar RAID, prefira o nível 0 do que outros, visto que o mesmo é feito para desempenho.

Mais abaixo vamos estudar alguns casos de empresas de tamanhos e necessidades diferentes, com todo o perfil de hardware utilizado.

É interessante também possuir um HD separado para os dados e para os logs do Squid. Se isso não for possível, ao menos uma partição separada é extremamente recomendado. Como normalmente tanto os dados quanto os logs fica abaixo do diretório /var, esse é o ponto de montagem para essa partição.

Sistemas de arquivo

Alguns sistemas operacionais são capazes de trabalhar com diversos sistemas de arquivos, tendo cada um suas características próprias, ora prezando por estabilidade, ora por desempenho.

Linux − reiserfs ou xfs

Windows 2000 − NTFS

DNS

O desempenho das resoluções DNS também é um ponto crítico. Em uma situação ideal, deveria existir um cache de DNS na mesma máquina ou em uma máquina muito próxima, para diminuir ao máximo o tempo de resolução dos nomes.

Múltiplas rotas

Em instalações como ISPs pode ser vantagem definir suas rotas manualmente. Já em empresas médias ou grandes que utilizam links de baixo custo, como ADSL, o balanceamento de carga nos links é uma ótima opção. Procure junto à documentação de seu sistema operacional como fazer isso.

Editando o squid.conf

Podemos também definir alguns parâmetros na configuração, de forma a obter o máximo do sistema.

```
cache_mem bytes
```

Nessa opção dizemos ao Squid quanta memória ele pode consumir. Em uma máquina exclusiva para o cache, 80% a 90% da memória total da máquina deve ser definida aqui.

Por exemplo, em uma máquina com 512MB de RAM:

```
cache_mem 410 MB
```

cache_swap_low percentage

Aqui se especifica o limite mínimo para substituição de um objeto. A substituição começa quando o swap em disco está acima do limite mínimo.

Defina algo como:

```
cache_swap_low 95
```

cache_swap_high porcentagem

Justamente o oposto da opção anterior. Aqui se define o limite máximo.

```
cache_swap_high 98
```

maximum_object_size bytes

A definição dessa propriedade deve ser analisada com critério, visto que limitamos aqui o tamanho máximo de um objeto em cache. Objetos maiores do que esse limite não são salvos em disco.

Para definir como configurar o tamanho máximo nessa opção, deve−se levar em consideração que um número grande implica em maior economia de banda e perda de performance no cache local, enquanto um número menor não ajuda muito em ganho de banda, mas melhora a velocidade em tempo de resposta. Recomenda−se a utilização de uma valor entre 4 e 16 MB.

```
maximum_object_size 16384 KB
```

maximum_object_size_in_memorybytes

Objetos maiores do que o tamanho definido aqui não são mantidos em memória. O tamanho deve ser grande o suficiente para armazenar objetos muito populares, mas pequeno demais para armazenar informações desnecessárias.

```
maximum_object_size_in_memory 20 KB
```

cache_dir Type Maxobjsize Directory−Name Mbytes Level−1 Level2

Configuramos nessa opção o tamanho máximo dos objetos dentro do diretório, o nome do diretório, quantos

MB armazenar e os níveis e sub−níveis.

É possível ter diversos diretórios de cachê, mas isso só vai fazer sentido se estiverem em HDs separadas. Caso a partição onde o seui Squid faz cache venha a encher, é possível criar um diretório de cache em outra partição, sem com isso obter ganhos de performance significativos.

```
cache_dir ufs /scsi2/cache 5000 16 256
```

Referências

http://www.pop−pb.rnp.br/proxy/tsld033.htm

Utilidades Públicas

Aqui estão alguns comandos que podem ser úteis.

Resetando o cache do squid

Pode ocorrer do squid travar alguma vez. Para tentar resolver isso, pare o squid e execute:

```
# squid −z
```

Reiniciando as configurações do squid

Se você mudou alguma ACL, atualizou a lista de sites ou qualquer coisa que exija refazer as regras do squid que está rodando, utilize:

```
# squid −k reconfigure
```

Entrando em modo Debug

Você pode modificar o Squid para modo Debug on the fly utilizando o seguinte comando:

```
# squid −k debug
```

O resultado do modo debug estará no arquivo cache.log, dentro do diretório de logs.

ATENÇÃO: A quantidade de logs gerada por esse modo é muito grande e irá causar lentidão no sistema. Não deixe essa opção habilitada por default.

Squid saindo com erro (Squid Parent: child process exited due to signal)

Quando ocorre um erro que impede a execução ou provoca a morte do squid, um aviso é enviado ao seu log assinalando o código do erro. Compilei aqui uma pequena tabela com alguns erros que encontrei e as soluções propostas.

Número	Verifique
6	Quantidade de memória disponível, espaço em disco, Bad Blocks no HD, problemas de DNS
9	O filesystem é read−only
11	Segmentation fault. Ou você econtrou um bug no Squid ou seu sistema (libs) está com problema
25	Veja se algum log tem mais de 2GB − access.log, cache.log ou store.log

Estudo de casos

Sempre é mais fácil aprender baseado em experiências práticas do que apenas em teoria. Vamos utilizar alguns exemplos reais aqui para vislumbrar o cenário em que nossas instalações irão se encaixar.

Simples, eficiente e muito útil

Em algumas localidades ainda não tem−se acesso a banda larga com facilidade. Ainda mais: Existem empresas que não querem ou não podem bancar o custo de uma conexão permanente. O cenário desse caso é o seguinte:

Empresa XYZ, do ramo de prestação de serviços, encontra−se localizada em uma região afastada, onde não pode ser atendida por meios convencionais de internet rápida. Os custos de uma conexão via satélite estão muito além do que a empresa está disposta a pagar. Seus 5 funcionários navegam na internet e usam e−mail somente via webmail. Sua missão é conectar essa empresa com baixo custo e eficiência.

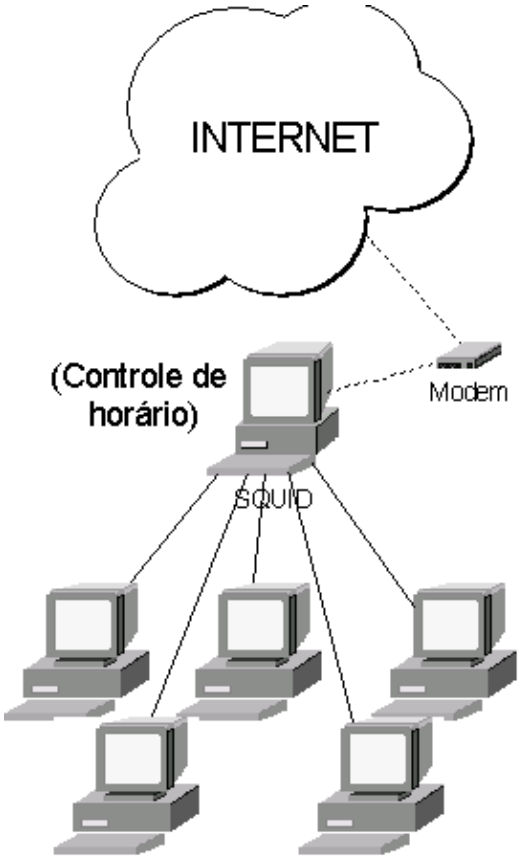


Diagrama da rede

Solução:

Adquirir um 486 DX4 100 ou maior (Qualquer hardware maior do que um Pentium 166 é desperdício) com 16 ou mais MB de RAM, com modem e placa de rede, além de uma HD em bom estado.

Instalar uma distribuição reduzida do Linux, com suporte a discagem sob demanda e configurar o Squid para restrição de acesso por horário, liberando o acesso a internet somente 1 ou 2 vezes por dia.

Justificativa:

Com um gasto em hardware bem pequeno, podemos conectar toda a empresa com um desempenho bom levando−se em conta se uma conexão discada. Além disso, a empresa garante que ninguém irá ficar conectado o dia inteiro através das restrições de horário de acesso impostas pelo Squid.

As pequenas dominam

Esse é o caso mais típico. Uma pequena empresa, normalmente de prestação de serviços ou comércio varejista, deseja ligar seu escritório ao resto do mundo pela internet. São cerca de 30 usuários ligados a uma rede cliente/servidor na plataforma Microsoft. Não existe uma verba muito grande para o projeto, logo é importante economizar o máximo em hardware e software para ganhar mais em serviço. Uma conexão ADSL já foi solicitada a empresa de telefonia e esse custo não é levado em conta no projeto.

Solução:

1− Adicionar uma segunda placa de rede ao servidor Microsoft.

2− Adquirir um appliance gateway/firewall (baixo custo) de uma das diversas marcas disponíveis no mercado e ligar sua interface WAN na conexão ADSL. Ligar sua interface LAN diretamente na nova placa de rede do servidor.

Instale o Squid no servidor Microsoft de acordo com as instruções dadas anteriormente e configure−o adequadamente.

Justificativa:

Apesar de muitas pessoas imediatamente ligarem o Squid ao Linux, não faz sentido não ter um projeto aprovado por causa dos custos da aquisição de uma nova máquina. Nem tampouco justifica−se a aquisição de sistemas caros e ineficientes para fazer de forma inadequada o que o Squid faz com perfeição. O appliance de firewall eu considero necessário porque conhecemos bem a sucessitibilidade da plataforma Microsoft a ataques. Mesmo que fosse uma plataforma 100% segura, ainda não deveríamos expor o servidor da rede local de forma tão aberta à internet.

Precoces

Por outro lado, existem empresas pequenas, talvez até micro, que têm uma visão de tecnologia mais à frente no mercado. Advocacias, contabilidades e empresas que trabalham com informações sigilosas em geral, têm consciência da necessidade de proteger os dados de seus clientes com firewalls seguros, sistemas de detecção de intrusos e etc.

Nosso cliente agora é uma advocacia com 8 usuários extremamente preocupada com o sigilo de seus dados e segurança de sua conexão com a internet. O desempenho da conexão não é tão importante quanto a auditoria dos sites acessados ou o bloqueio de eventuais vírus.

Solução:

1− Adquirir um servidor novo para instalação do firewall

Proceder instalando o Linux da forma mais segura possível, de preferência aplicando patches no kernel e instalando sistemas de auditoria interna. Um sistema de detecção de intrusos também é essencial. Feito isso, instalar o Squid com método de autenticação, viralator e restrição de conteúdo. Lembre−se de que nesse caso, talvez seja interessante bloquear acesso também a webmails.

Instalar também o sarg e gerar relatórios diários de utilização. Utilizar−se da facilidade de logrotate e faze backup diário dos logs.

Justificativa:

A preocupação da empresa com o sigilo de seus dados e de seus clientes vale o investimento em uma nova máquina, que poderá fornece−lhes todas as informações necessárias para auditoria e solução de possíveis falhas.

Arroz com feijão

Esse é o caso mais comum de todos. Creio que 80% das instalações que já fiz seguem esse padrão. No cenário temos uma ou várias empresas, de porte de pequeno a grande em uma mesma localidade física e com apenas um link ligando−as à internet. Já pudemos participar de implantações onde o link variou de um frame−relay de 64Kbps segurando uma única empresa até conexões de fibra óptica de 2Mbits onde várias empresas e usuários de um condomínio ou prédio faziam uso dessa para acesso a internet em geral. O proxy deve ser transparente e o único objetivo do Squid é dar ganho de velocidade e economia do link.

Solução:

1− Adquirir um servidor de qualidade, analisando a necessidade de hardware da instalação

Configurar o firewall utilizado e o Squid para trabalhar de forma transparente. Recomendo uso do Linux ou do FreeBSD, de acordo com sua familiaridade com esses sistemas. Procure alterar parâmetros de memória e espaço em disco utilizado. Talvez seja bom reavaliar a instalação após 1 ou 2 meses, procurando uma sintonia fina de parâmetros.

Justificativa:

Em uma relação de custo e benefício de médio e longo prazo, podemos perceber que é mais barato instalar um servidor de cache do que aumentar um link. Isso é especialmente verdade quando falamos de conexão de qualidade. Como sempre, tanto o Linux como o FreeBSD não só fazem o serviço por um valor quase irrisório, como também o fazem com perfeição.

15.5 Uma empresa sadia

Com certeza você um dia irá se deparar com um projeto de maior profundidade, com complicadores e detalhes chatos. O caso apresentado aqui é de uma empresa da área da saúde, que desejava ao mesmo tempo ter estabilidade, desempenho, auditoria e monitoramento de usuários, além de níveis de acesso e uma exigência do presidente que podemos dizer ser, no mínimo, pitoresca: Um link só para ele, sem log , sem auditoria e sem perguntas.

Para não dizer que só temos problemas, a verba era bem gorda para a implantação e razoável para a manutenção.

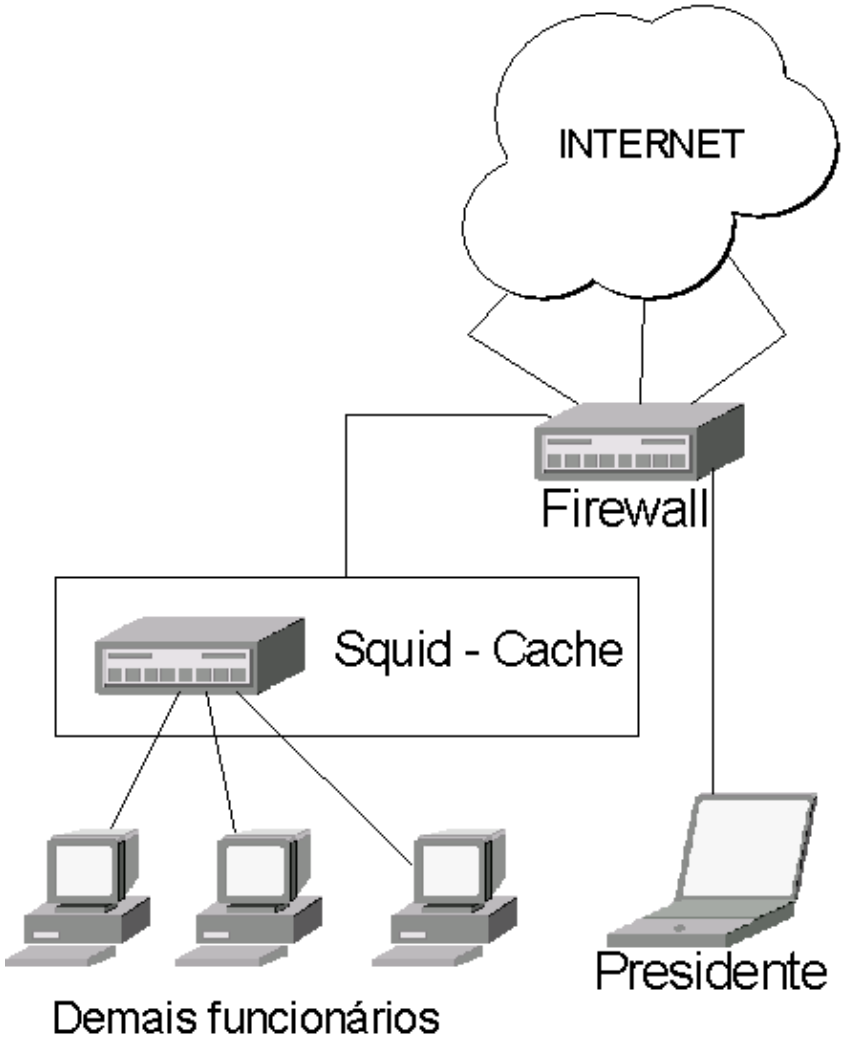


Diagrama da rede

Solução:

1− Adquirir 3 links ADSL de velocidades 512Kbps (x2) e 2Mbits (1x)

2− Adquirir 2 servidores (um deles com 5 placas de rede)

Instalar Linux em ambos os servidores. 1 deles será o Cache e o outro o firewall.

No firewall deve ser configurado balanceamento de carga entre os 2 ADSLs de 512Kbps (WAN1 e WAN2), enquanto o de 2Mbits deve ficar isolado (WAN3). Esse servidor deve ter regras rígidas de firewall e de roteamento interno, de forma que apenas o servidor de cache tenha acesso a sua interface de rede LAN1 e apenas a máquina do presidente tenha acesso à LAN2. A melhor solução seria restrição por MAC Address no firewall.

No cache, deve−se instalar um sistema de autenticação, restrição de horário, restrição de sites e geração de

logs.

Justificativa:

Com o firewall bem configurado e com roteamento e balanceamento de carga definidos, impedimos que algum usuário mais esperto tente burlar o cache. Da mesma fora permitimos que o presidente acesse a web por seu link exclusivo sem cache e sem log.

Todos os usuários, por sua vez são muito bem controlados em tudo o que fazem. Garante−se assim todos os requisitos exigidos pelo cliente.

Matriz e filial

Não são raros os casos de empresas que dispõe de link com a internet apenas em sua matriz e todas as suas filiais interligadas a ela por frame−relay. Administradores que não conhecem (ou não conheciam) o conceito de caching, perdem uma imensa quantidade de banda com navegação de suas filiais na internet. Alguns administradores tentam evitar esse problema colocando um cache na matriz. O resultado é muito bom, economizando a largura de banda necessária para outros serviços. No entanto a comunicação entre as filiais e matriz continua prejudicada devido à navegação. Como resolver isso?

Solução:

1− Adquirir um servidor para cada filial

Em cada filial será instalado um servidor cache utilizando o modo transparente e com configurações de hierarquia, onde todas as filiais serão filhas da Matriz.

Justificativa:

Com essa solução, começamos diminuindo o tráfego até a matriz com o cache local. Mesmo que um determinado objeto não esteja na memória do servidor da filial, o mesmo será verificado no servidor da matriz, economizando a saída até a internet. Como todas as filiais estão passando pelo mesmo servidor final, provavelmente a economia será muito grande em termos de banda IP.

Cache aéreo

Em diversas cidades estão surgindo os provedores wireless. Impulsionados por uma tecnologia barata, de simples implementação e manutenção, vários condomínios residenciais e comerciais estão recebendo seus links desse tipo de provedor. Como todo ISP sabe, o grande custo é o link com a internet. Alguns milhares de reais são gastos mensalmente para manter um link apenas rápido o suficiente para a demanda. Sendo assim, todo e qualquer esforço é válido para evitar o upgrade de link.

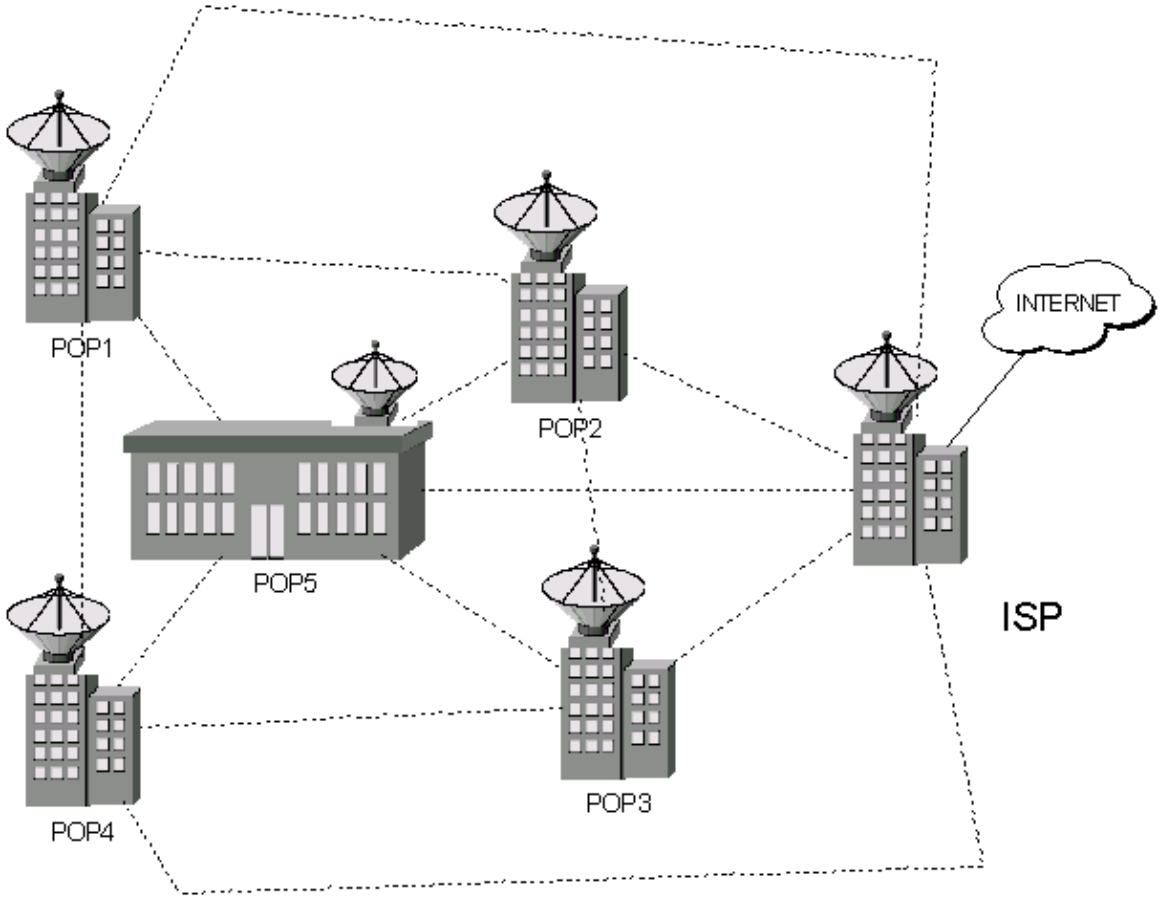


Diagrama da rede

Nosso cenário aqui é exatamente esse. Um ISP wireless com um backbone central e diversos pontos de presença.

Solução:

1− Adquirir um servidor para cada POP3

Todos os caches serão configurados no modo transparente e o mais otimizados possível. Uma hierarquia será montada, de preferência no modo horizontal, ou grupo. O modelo árvore pode gerar tráfego desnecessário até a central.

Todos os servidores de uma determinada nuvem devem se consultar mutuamente, de forma a manter o máximo de tráfego em uma única região geográfica. Dentro do ISP, um outro proxy transparente, muito bem configurado irá fazer ainda uma última verificação na memória antes de finalmente buscar uma página na web.

Justificativa:

Haverá uma significativa economia de banda IP, além de conseqüente diminuição de tráfego nos APs4 e uma economia muito grande para o provedor. Além de fornecer aos clientes um serviço de excepcional qualidade

por um valor bem viável.

ISP

Simplesmente não existem ISPs que não queiram dar uma melhor qualidade de serviço para seus clientes e diminuir seus custos com link. Nossa proposta é de resolver essas duas questões de uma única vez utilizando o squid. Deve−se estudar com cautela o seu caso para que ele se enquadrar a melhor solução.

Nesse caso não pretendo dar uma solução como nos anteriores. A única exigência é utilizar proxy transparente. Dependendo do porte do seu provedor, a solução pode ser colocada em um único servidor com uma HD IDE ou então em um cluster de alta disponibilidade com discos SCSI e controladora RAID.

Pense sempre no custo x benefício. Um pequeno provedor não pode comprar um servidor de milhares de reais, da mesma forma que um grande provedor que coloca uma máquina de baixa qualidade corre o risco de ter degradação de desempenho em relação ao uso sem cache.

Mantenha em sua mente que a parte mais cara do provedor em termos de infra−estrutura é a banda IP. Talvez um servidor de dezenas de milhares de reais seja extremamente barato para um grande provedor, levando−se em conta a economia de banda gerada.

Em grandes instalações, procure entrar em contato com seu fornecedor de banda IP e veja a possibilidade de interligar seu Squid com um dos proxies internos da telecom. Isso não irá gerar economia de banda local, mas dará um ganho de velocidade e qualidade.

Tente também conversar com administradores de outras grandes instalações, analisando a possibilidade de interligar os ISPs por um link dedicado. Isso será mais barato que uma banda IP e, além das rotas, você ainda pode configurar seus proxies para utilizar ACL por domínio, com a criação de um cache exclusivo para o outro ISP.

Examinando o Squid.conf

A partir de agora, vamos explicar passo a passo as tags de configuração do squid.conf.

Lembre−se de que alterações bem feitas e pensadas podem trazer um grande ganho para a performance de seu cache, enquanto um erro de configuração pode impedir seu Squid de trabalhar ou remover muitas de suas funcionalidades.

Altere as opções com cautela e certifique−se de que realmente necessita fazer a mudança que planeja.

Tags da seção Network

Essa seção explica todos os parâmetros de endereços de redes relevantes para uma instalação do Squid.

http_port

O número da porta onde o Squid irá ouvir as requisições dos clientes. O padrão é 3128. Essa opção será ignorada quando o squid é iniciado com a opção "−a" na linha de comando

Você pode espeficicar múltiplas portas, em qualquer uma das três formas: somente a porta, por hostname e porta ou IP e porta. Se você espeficiar um hostname ou endereço IP, então o Squid irá ouvir naquele endereço especificado.

http_port porta

http_port ip:porta

hostname: porta

1.2.3.4 : porta

icp_port

Especifica o número da porta na qual o squid irá enviar e receber solicitações ICP de outros Cache Servers. Para desabilitar, basta colocar um 0. Padrão: 3130

Como já dito anteriormente, o ICP é usando para comunicação entre caches, provendo as funcionalidades necessárias para troca de informações sobre objetos armazenados.

icp_port porta

htcp_port

Especifica o número da porta através do qual o Squid irá receber e enviar requisições HTCP de e para caches vizinhos. Para desabilitar, colocar 0. O padrão é 4827.

Specify the port number through which Squid sends and receives HTCP queries to and from neighbor caches. To disable "0" is used (default = 4827).

htcp_port porta

mcast_groups

Especifica uma lista de grupos multicast, no qual seu servidor pode juntar−se para receber requisições ICP. Padrão = none

mcast_groups Endereço_IP

tcp_outgoing_address

É usado para conexões feitas em servidores remotos. Também é usado para comunicar−se com outros caches durante o uso de HTCP ou CARP. Normalmente não deve−se especificar tcp_outgoing_address. A melhor opção é deixar o sistema operacional escolher um endereço. Padrão: 255.255.255.255

tcp_outgoing_address Endereço_IP

udp_incoming_address

É usado pelo socket ICP para receber pacotes de outros caches. Padrão: 0.0.0.0

udp_incoming_address Endereço_IP

udp_outgoing_address

É usado pelo socket ICP para enviar pacotes a outros caches. Padrão: 255.255.255.255

udp_outgoing_address Endereço_IP

Tags da seção Peer cache servers e Squid hierarchy

As tags dessa seção são relevantes quando rodando o Squid em uma rede com hierarquia.

cache_peer

Especifica outros caches na hierarquia. A opção cache_peer é dividida em 5 campos. O primeiro campo é o IP ou nome do servidor do cache que será pesquisado. O segundo indica o tipo de relacionamento. No terceiro configura−se a porta HTTP do servidor destino. No quarto campo configura−se a porta de requisição ICP e, finalmente, o quinto campo pode conter zero ou algumas palavras−chave.

cache_peer hostname tipo porta_http porta_icp [opções]

Parâmetros Descrição

Hostname Hostname (FQDN) ou endereço IP do cache a ser pesquisado.

tipo Aqui especifica−se a hierarquia de cache definida. Opção importante para escolha de regras de vizinhança.

Opções:

- parent
- sibling
- multicast
- porta_http O número da porta onde o cache ouve as requisições http.
- porta_icp O número da porta onde o cache ouve as requisições http.

Opções	**Descrição**
proxy−only	Especifica que os objetos desse servidor não devem ser salvos localmente
Weight=n	Especifica o peso de um "pai". Deve ser um valor inteiro, sendo que o padrão é 1. Servidores com um peso maior tem preferência
ttl=n	Especifica o tempo de vida de um multicast
no−query	Essa opção será utilizando quando fazendo requisições a caches que não aceitam ou não suportam ICP. Caso utilize essa opção, configure o quarto campo como 0
default	Se esse cache será usado como uma última opção e ele não está configurado para trabalhar com ICP, entãp utilize essa opção. Ele não será o padrão, mas sim a última opção, apesar do que indica o nome da tag
round−robin	Define uma série de "pais" que podem ser usados baseados em algorítimo round−robin.
multicast−responder	Indica que o servidor indicado é membro de um grupo de multicast.
closest−only	Indica que, para uma resposta ICP_OP_MISS, nós somente iremos passar CLOSEST_PARENT_MISS e nunca FIRST_PARENT_MISS.
no−digest	Não faz requisições tipo digest para esse vizinho.
no−netdb−exchange	Desabilita requisições ICMP RTT desse vizinho
no−delay	Evita que esse vizinho seja influenciado por uma delay pool.
login=usuário:senha	Caso esse servidor exija autenticação.
connect−timeout=nn	Especifica o time out para essa conexão.
digest−url=url	Diz ao Squid para buscar o resumo do cache utilizando essa URL.
cache_peer_domain	Limita o domínio para qual cada vizinho será requisitado. É usado para enviar requisições para caches diferentes dependendo do domínio.

- Colocar um '!' antes do domíno significa que o cache irá armazenar o que não for para tal.
- Pode−se colocar tantos domínios quanto necessário por cache, tanto na mesma linha como em linhas separadas.
- Quando múltiplos domínios são dados para um único cache, o primeiro domínio é aplicado.

- Cache hosts sem domínio irão aceitar todos os pedidos

  cache_peer_domain cache_host domínio [domínio]

neighbor_type_domain

Modifica o tipo do servidor vizinho dependendo do domínio. Você pode tratar domínios de forma diferente quando um servidor padrão é usado na tah cache_peer.

neighbor_type_domain parent|sibling domínio [domínio]

icp_query_timeout

Aqui pode−se definir manualmente o timeout de uma requisição ICP. Visto que o Squid irá automaticamente determinar um valor ideal baseado em requisições recentes, é bom não alterar essa opção.

icp_query_timeout milisegundos

maximum_icp_query_timeout

Tempo máximo de expiração de uma requisição ICP. A resposta não será mais esperada depois desse tempo.

maximum_icp_query_timeout milisegundos

mcast_icp_query_timeout

Normalmente o Squid envia pacotes de teste para os endereços multicast para determinar quais servidores estão na escuta. Essa opção determina quanto tempo o Squid irá esperar por uma resposta.

Como o Squid fica aguardando resposta, não coloque um valor muito alto. O padrão está OK. 2000 ms.

mcast_icp_query_timeout milisegundos

dead_peer_timeout

Controla quanto tempo o Squid leva para declarar um servidor como morto. Se nenhuma requisição ICP for respondida nesse tempo, o Squid continuará mandando requisições ICP, mas não esperará por resposta. O servidor será novamente marcado como vivo depois que uma determinada seqüência de respostas for enviada. Padrão de 10 segundos.

dead_peer_timeout segundos

hierarchy_stoplist

Uma lista de palavras que, encontradas na URL, farão com que o objeto seja manipulado automaticamente por esse cache.

hierarchy_stoplist palavras

no_cache

Uma lista de elementos de uma ACL, onde, se encontrados, impedem o objeto de ser cacheado.

no_cache deny|allownomeacl

Tags da seção Cache size

Descreve os parâmetros relacionados ao tamanho da memória utilizada pelo cache, assim como a política de rotatividade na memória. O Squid suporte mais que uma política de rotatividade de memória.

cache_mem

Especifica o número ideal de memória usado para:

- Objetos em transito
- Objetos "quentes"
- Objetos com negativa de cache

Os tamanho dos dados para esses objetos são definidos em blocos de 4 KB. Esse parâmetro especifica o limite ideal para os blocos alocados. Objetos em transito tem prioridade sobre os outros. Quando espaço adicional é necessário para novos dados, objetos "quentes" e com negativa de cache são liberados. Padrão de 8MB.

cache_mem total MB

cache_swap_low

Aqui se especifica o limite mínimo para substituição de um objeto. A substituição começa quando o swap em disco está acima do limite mínimo. Padrão de 90.

cache_swap_low porcentagem

cache_swap_high

Justamente o oposto da opção anterior. Aqui se define o limite máximo. Padrão de 95.

cache_swap_high porcentagem

maximum_object_size

A definição dessa propriedade deve ser analisada com critério, visto que limitamos aqui o tamanho máximo de um objeto em cache. Objetos maiores do que esse limite não são salvos em disco.

Para definir como configurar o tamanho máximo nessa opção, deve−se levar em consideração que um número grande implica em maior economia de banda e perda de performance no cache local, enquanto um número menor não ajuda muito em ganho de banda, mas melhora a velocidade em tempo de resposta. Recomenda−se a utilização de uma valor entre 4 e 16 MB. No padrão será utilizado 4096 kB.

maximum_object_size bytes

minimum_object_size

Objetos menores do que esse valor não serão armazenado em cache. O valor padrão é 0, o que significa que todos os objetos serão armazenados.

minimum_object_size bytes

maximum_object_size_in_memory

A definição dessa propriedade deve ser analisada com critério, visto que limitamos aqui o tamanho máximo de um objeto em cache. Objetos maiores do que esse limite não são salvos em disco.

Para definir como configurar o tamanho máximo nessa opção, deve−se levar em consideração que um número grande implica em maior economia de banda e perda de performance no cache local, enquanto um número menor não ajuda muito em ganho de banda, mas melhora a velocidade em tempo de resposta. Recomenda−se a utilização de uma valor entre 4 e 16 MB.

maximum_object_size_in_memory bytes

ipcache_size

Especifica o tamanho do cache de ip.Padrão de 1024.

ipcache_size número_entradas

ipcache_low

Especifica o número mínimo de IPs cacheados. Padrão de 90.

ipcache_low porcentagem

ipcache_high

Especifica o número máximo de IPs cacheados. Padrão de 95.

ipcache_high porcentagem

fqdncache_size

Especifica o número máximo de FQDNs cacheados. Padrão de 1024.

fqdncache_size número_entradas

cache_replacement_policy

Define qual objeto será mantido na memória e qual será removido para criar espaço para novos objetos.

Opção	**Descrição**

LRU	A opção padrão utilizada pelo Squid. Mantém em cache objetos referenciados a recentemente, ou seja, começa removendo do cache o objeto que foi referenciado a mais tempo.
heap GDSF	Tem a filosofia de mantém em cache objetos menores, referenciados mais vezes, gerando uma maior possibilidade de fornecer um hit.
heap LFUDA	Mantém os objetos mais populares em cache, independente de seu tamanho.
heap LRU	Política LRU acrescida do uso de pilhas.

cache_replacement_policy política

memory_replacement_policy

Determina quais objetos são removidos da memória quando é preciso liberar espaço. Segue as mesmas políticas do cache_replacement_policy

memory_replacement_policy política

Tags da seção Log file path names and cache directories

Descreve os parâmetros para configuração dos diretórios de cache e log em disco. Os arquivos de log são importantes não só para troubleshooting, mas também geração de relatório e observação de anomalias.

É recomendável que você utilize−se de uma política de rotacionamento de log, como o log−rotate.

cache_dir

Diretório onde serão armazenados os objetos. É possível criar−se vários diretórios de cache, mas isso só irá fazer sentido se os mesmo forem em partições (ganho de espaço) ou discos (ganho de velocidade) separados.

tipo Especifica o tipo de arquivo q ser criado. Utilize o ufs. A opção aufs deve ser utilizada quando em um Linux ou Solaris com I/O Assíncrono.

tamanho_máx_obj Refere−se ao tamanho máximo do objeto que será armazenado nesse diretório.

nome_diretório É o raiz do diretório de cache. Caso esteja utilizando um disco separado para o cache, será o ponto de montagem. O diretório já deve existir previamente e o usuário do Squid deve ter direito a escrita nele.

Mbytes Quantidade de espaço em disco ocupado por esse diretório. Definido em MB.

nível−1 Número de subdiretórios de primeiro nível criados sob o diretório principal.

nível−2 Número de subdiretórios de segundo nível que será criado abaixo de cada subdiretório de primeiro nível.

cache_dir tipo tamanho_máx_obj nome_diretório Mbytes nível−1 nível2 [..]

cache_access_log

Especifica o caminho para o arquivo de logs de acesso, o qual guarda todas as requisições e atividades de clientes. Os detalhes do log podem ser customizados como log_mime_hdrs, log_fqdnm client_netmask e emulate_httpd_log. Padrão: /usr/local/squid/logs/access.log.

cache_access_log path_diretório/nome_arquivo

cache_log

Configura o caminho para o log de cache. Esse arquivo irá conter informações gerais sobre o comportamento do Squid. Padrão: /usr/local/squid/logs/cache.log.

cache_log path_diretório/nome_arquivo

cache_store_log

Diz qual o caminho do log de armazenamento. Esse arquivo contém detalhes sobre o processo de armazenamento em disco, podendo fornecer informações como quais arquivos foram removidos do cache, quais foram mantidos e por quanto tempo. Padrão: /usr/local/squid/logs/store.log.

cache_store_log path_diretório/nome_arquivo

cache_swap_log

Caminho para o arquivo swap.log. Esse arquivo contém metadados sobre objetos salvos em disco, podendo ser utilizado para dar um "rebuld" no cache durante a inicialização. Normalmente ele fica armazenado no primeiro diretório de cache, mas pode ter o caminho alterado com essa opção. Esse arquivo não pode ser rotacionado.

Se você tem mais de um cache_dir, então o seu arquivo de log de swap terá nomes como:

- cache_swap_log.00
- cache_swap_log.01
- cache_swap_log.02

cache_ swap _log path_diretório/nome_arquivo

emulate_httpd_log on|off

O Squid tem a habilidade de emular o log de servidores web. Para utilizar essa opção, basta configurar com "on". Se você não tem nenhuma aplicação específica para utilização do log em formato web, sugiro que mantenha no padrão do Squid, visto que será mais simples encontrar ferramentas de analise de logs nesse padrão.

emulate_httpd_log on|off

log_ip_on_direct

Ativa/Desativa a opção de loggin para um IP destino em uma hierarquia quando o cache direciona a requisição de um servidor origem.

log_ip_on_direct on|off

mime_table

Configura a tabela MIME do Squid. Esse arquivo irá conter os tipos MIME suportados pelo Squid. Padrão: /usr/local/squid/etc/mime.conf.

mime_table path_diretório/nome_arquivo

log_mime_hdrs on|off

Grava tanto as requisições quanto as respostas MIME no cabeçalho de cada transação HTTP. Os cabeçalhos irão aparecer em 2 partes diferentes no access.log.

log_mime_hdrs on|off

user agent_log

Para utilizar essa opção, o Squid precisa ter sido compilado com a opção "−−enable−useragent_log". Com isso será possível agravar em um log o User−Agent de todas as requisições http. Desabilitado por padrão.

useragent_log path_diretório/nome_arquivo

referer_log

Também necessita que o Squid tenha sido compilado com uma opção extra: "−−enable−referer_log". Esse log irá guardar todas as referências das requisições HTTP. Desabilitado por padrão.

referer_log path_diretório/nome_arquivo

pid_filename

Especifica em qual arquivo será arquivado o PID dos processos do Squid. Padrão: /usr/local/squid/logs/squid.pid.

pid_filename path_diretório/nome_arquivo

debug_options

Como os logs são configurados por nível, podemos configurar o tanto de informações que o Squid irá gerar para nossa análise. Recomendo que utilize o padrão, exceto se estiver tendo algum problema que não possa ser facilmente diagnosticado. Quanto menor o nível de log, menos informações serão geradas. Usando a palavra ALL, podemos configurar o nível de log em todos de uma única vez. Padrão: ALL, 1.

debug_options seção,nível

log_fqdn

Pode ser configurado como ON, se você deseja logar o FQDN no access.log. Por padrão está desabilitado.

log_fqdn on|off

client_netmask

A máscara de rede para o endereço de clientes e saída do cachemgr. Utilize o padrão como melhor opção.Padrão: 255.255.255.255.

client_netmask máscara_rede

Tags da seção Support for External functions

Solicita certas funções externas que não são parte do binário do Squid. Esse executáveis normalmente são relacionados a DNS, ftp, redirecionamento e autenticação.

Eles são chamados pelo Squid através de fork() ou exec() padrão. O número de forks filhos será especificados para cada processo externo.

Parâmetros relevantes para essa seção:

ftp_user

Essa tag é utilizada se você deseja que o login anônimo seja mais informativo. Coloque alguma informação significativa como proxy@seudominio.com.br. Padrão: Squid@.

ftp_user nome_usuário

ftp_list_width

O tamanho da largura da lista dos arquivos do ftp. Um número muito pequeno irá cortar nomes de arquivos grnades quando navegando em sites web. Padrão: 32

ftp_list_width número

ftp_passive

Se o seu firewall não permite que o Squid use conexões passivas, desligue essa opção.

ftp_passive on|off

cache_dns_program

Define−se aqui o caminho para o executável do dns lookup. Essa opção só está disponível se o Squid for compilado com a opção −−disable−internal−dns.

O programa de dns externo usa as bibliotecas de resolução, provendo um cliente de dns muito mais amadurecido e confiável. Caso não haja nada de estranho com sua resolução de DNS do Squid, mantenha o resolver interno.

cache_dns_program programa

dns_children

Número de processos simultâneos para o serviço de DNS. Para servidores com grande load, pelo menos 10 filhos devem ser iniciados. O máximos fica em 32 filhos, sendo o padrão 5. Novamente é preciso ter compilado o Squid especialmente para suporte a DNS externo. Quanto mais rápida a resolução DNS, melhor o desempenho geral do sistema. Tendo isso em mente, utilize 32 processos filhos.

dns_children número

dns_retransmit_interval

Tempo inicial que o DNS aguarda para retransmitir uma solicitação. O intervalo dobra cada vez que todos os DNS configurados são tentados.

dns_retransmit_interval segundos

dns_timeout

Timeout para requisições DNS. Se não houver resposta depois desse tempo, todos os DNS configurados para esse domínio são considerados indisponíveis.Padrão de 5 minutos.

dns_timeout minutos

dns_defnames

Normalmente o servidor de dns desabilita a opção de resolução RES_DEFNAMES. Isso impede que caches em uma hierarquia resolvam nomes de hosts localmente. Para utilizar essa opção, não esqueça de habilitar na hora da compilação.

dns_defnames on|off

dns_nameservers

Pode ser usada para especificar uma lista de servidores DNS no lugar no /etc/resolv.conf

dns_nameservers Endereço_IP

unlinkd_program

Especifica o caminho do programa unlinkd. Isso não é necessário se você estiver usando I/O assíncrono. Padrão: /usr/local/squid/libexec/squid/unlinkd.

unlinkd_program path_diretório/nome_programa

diskd_program

Especifica a localização do diskd.

diskd_program path_diretório/nome_programa

pinger_program

Define o caminho do executável pinger.

pinger_program path_diretório/nome_programa

redirect_program

Diz qual o caminho do redirecionador de URL. Existem diversas aplicações que poderão ser utilizadas aqui.

redirect_program path_diretório/nome_programa

redirect_children

Número de processos filhos para o programa de redirect.

redirect_children número

redirect_rewrites_host_header

Por padrão o Squid reescreve o header de host em requisições redirecionadas. Se você está rodando como proxy reverso, isso pode não ser desejado.

redirect_rewrites_host_headeron|off

redirector_access

Se definido, essa lista de acesso especifica quais requisições são enviadas para o processo de redirect. Por padrão, todas o são.

redirector_access allow|deny

authenticate_program

Especifica o comando do autenticador externo. Esse programa lê uma linha contendo: "usuário senha" e devolve um OK ou ERR. Para utilizar o autenticador é preciso ter uma ACL relacionada.

authenticate_program path_diretório/nome_programa path_diretório/arquivo_senhas

authenticate_children

Número de processos filhos do autenticador. Padrão de 5.

authenticate_children número

authenticate_ttl

Especifica o tempo de vida para uma autenticação bem sucedida permanecer em cache. Se uma combinação inválida de nome de usuário e senha é fornecida, o usuário é removido do cache e uma revalidação é exigida. Padrào de 3600 segundos.

authenticate_ttl segundos

authenticate_ip_ttl

Com essa opção você poderá especificar por quanto tempo a autenticação persistirá para um determinado IP. Se uma requisição usando a mesma autenticação da conexão já efetuada for utilizada em outra máquina, ambas terão acesso bloqueado e será exigida uma nova autenticação. Se você tem usuários com uma conexão discada conectando em seu Proxy remotamente, é recomendável que não tenha um número maior do que 60 segundos, visto que isso o impediria de conectar−se novamente durante esse tempo se a linha dele caísse. O padrão é de 0 segundos.

authenticate_ip_ttl segundos

authenticate_ip_ttl_is_strict

Essa opção faz com que a autenticação seja um pouco mais rígida. Ela impede que qualquer outra conexão seja feita com outros endereços IP enquanto o tempo de vida especificado anteriormente não expirar. Essa opção está ativada por padrão.

authenticate_ip_ttl_is_stricton|off

Tags da seção para tunning do Squid

Essa seção descreve importantes parâmetros para determinar a performance do Squid.

wais_relay_host / wais_relay_port

Define o servidor de relacionamento WAIS

wais_relay_host host

wais_relay_port porta

request_header_max_size

Especifica o tamanho máximo de um cabeçalho de uma requisição http. Como sabe−se que um cabeçalho HTTP deve ser pequeno (por volta de 512 bytes), limitar o tamanho do mesmo pode ser interessante no uso de proxy reverso, criando uma barreira a mais para ataques do tipo buffer overflow e denial of service. Padrão de 10K.

request_header_max_size kbytes

request_body_max_size

Especifica o tamanho máximo para o corpo de uma requisição HTTP. Ou seja, o tamanho máximo de um PUT ou POST. Essa opção pode ser interessante para empresas que queiram garantir que seus usuários não farão grandes uploads à partir da empresa. Padrão de 1MB..

request_body_max_size kbytes

reply_body_max_size

Tamanho máximo do corpo de um reply. Isso é útil para impedir que seus usuários baixem arquivos grandes. Padrão de 0.

reply_body_max_size kbytes

refresh_pattern

Essa opção deve ser usada com extremo cuidado. Se você não tiver nenhuma aplicação que exija explicitamente alterar essa TAG, sugiro que deixe−a inalterada. Um valor inadequado aqui fará com que seus usuários simplesmente não consigam mais acessar aplicações dinâmicas na web. Não seja levado pela idéia de que impedir os usuários de ficar dando reload em uma página irá economizar sua banda, pois a dor de cabeça gerada será muito mais cara do que sua banda.

Parâmetros	Descrição
mín	Tempo mínimo, em minutos, que um objeto sem um tempo de expiração explicitamente configurado será considerado válido. Utilize, impreterivelmente 0.
porcentagem	É a porcentagem da idade dos objetos, desde a última modificação, no qual esse será considerado válido, desde que não tenha um valor de expiração configurado.
máx	É o tempo máximo, em minutos, que um objetos sem um tempo de expiração explicitamente configurado será considerado válido.

Opções	Descrição
override−expire	Reforça o tempo mínimo de expiração de um objeto, ainda que o mesmo tenha sido enviado no cabeçalho.
override−lastmod	Reforça o tempo mínimo, ainda que o objeto tenha sido modificado recentemente.
reload−into−ims	Modifica solicitações do tipo "sem−cache" ou "reload" para "Se−modificado−desde−requisição"
ignore−reload	Simplesmente ignora as requisições "sem−cache" e "reload".

refresh_pattern [−i] regex mín porcentagem máx [opções]

reference_age

Como já discutido, o Squid atualiza sua memória baseado em políticas, normalmente removendo primeiro objetos mais antigos ou menos populares. Apesar disso ser feito dinamicamente, podemos configurar valores manualmente nessa opção, configurando o tempo máximo de permanência em memória. O valor padrão é de 1

ano.

reference_age tempo

quick_abort_min / quick_abort_max / quick_abort_pct

O cache pode ser configurado para continuar com o download de requisições abortadas. Ao mesmo tempo que isso pode ser indesejado em redes pequenas e com conexão lenta, pode ser útil em grandes instalações, onde quase certamente um outro usuário irá requisitar o mesmo objeto.

Quando o usuário aborta um download, o Squid verifica o valor da opção quick_abort e a quantidade de dados baixados até o momento. Se o transferido for menor do que o especificado, ele irá finalizar o download. Padrão: 16 KB

Se o transferido tiver mais do que o quick_abort_max, ele irá abortar a transferência. Padrão: 16 KB

Se uma porcentagem maior do que a configurada em quick_abort_pct tiver sido baixada, ele finaliza o download. Padrão de 95%.

quick_abort_min kbytes

quick_abort_max kbytes

quick_abort_pct porcentagem

negative_ttl

Tempo de vida para requisições falhas. Certos tipos de erros (como conexão recusada ou página não encontrada) são marcados como "sem−cache" por um determinado tempo. Padrão de 5 minutos.

negative_ttl tempo

positive_dns_ttl

Tempo de vida para resultados bem sucedidos de resolução DNS. Se você realmente precisar alterar esse valor, não deixe inferior a 1 minuto. Padrão de 6 horas.

positive_dns_ttl tempo

negative_dns_ttl

Tempo de vida de resoluções falhas de DNS.

negative_dns_ttl tempo

range_offset_limit

Configura um limite superior de até onde deverá ir a abrangência de uma requisição de arquivo em um pré−download. Se passar desse limite, o Squid encaminha a requisição como está e não cacheia o resultado.

range_offset_limit bytes

Tags da seção Timeouts

Parâmetros de time out podem ser baseados em tempo de conexão, conexão com host, por site ou domínio, por tipo de requisição, etc. Time outs bem configurados são essenciais para otimizar a performance do Squid. Os principais parâmetros estão listados abaixo.

connect_timeout

O tempo de espera que o Squid aguarda pela resposta do servidor de origem. Se esse tempo for excedido, o Squid responde com uma mensagem de "Connection timed out". Padrão de 120 segundos.

connect_timeout segundos

peer_connect_timeout

Especifica quanto tempo deverá ser aguardada uma resposta de um cache vizinho para conexões TCP. Diferentes limites podem ser configurados para vizinhos distintos. Padrão de 30 segundos.

peer_connect_timeout segundos

site select_timeout

Define o tempo de expiração para URN em seleção de múltiplas URLs. URN é um protocolo desenvolvido para resolução de nomes independente de localização. Padrão de 4 segundos.

siteselect_timeout segundos

read_timeout

Essa opção é usada em conexões server−side. Após cada leitura bem sucedida, o time out será aumentado nesse valor. Se nenhum dado for lido após esse tempo, a requisição é abortada e logada como ERR_READ_TIMEOUT. Padrão de 15 minutos.

read_timeout tempo

request_timeout

Diz ao Squid quanto tempo esperar após uma conexão HTTP ser aberta. Para conexões persistentes, o Squid irá aguardar esse tempo após o fim da requisição anterior. Default de 30 segundos.

request_timeout segundos

client_lifetime

Tempo máximo que um cliente poderá ficar conectado ao processo de cache. Entenda−se cliente como browser. Isso protege o cache de ter muitos sockets em estado CLOSE_WAIT devido a clientes que desconectam sem utilizar o procedimento adequado. Padrão de 1 dia.

client_lifetime tempo

half_closed_clients

Alguns clientes podem parar o envio de pacotes TCP enquanto deixam o recebimento em aberto. Algumas vezes o Squid não consegue diferenciar conexões TCP totalmente fechadas e parcialmente fechadas. Por padrão, conexões parcialmente fechadas são mantidas abertas até que haja um erro de leitura ou escrita no socket. Mudando essa opção para off fará com que o Squid imediatamente feche a conexão quando a leitura do socket retornar "sem mais dados para leitura".

half_closed_clients on|off

pconn_timeout

Aqui configura−se o timeout para conexões persistentes. Depois do tempo de inatividade determinado aqui, o Squid encerra as conexões persistentes. Caso você configure essa opção para menos de 10 segundos, a funcionalidade estará desabilitada. Padrão de 120 segundos.

pconn_timeout segundos

ident_timeout

Tempo máximo para para aguardar requisições IDENT. Se esse valor estiver muito alto e a opção ident_lookup ativada, existe a possibilidade de sujeitar−se a uma negação de serviço, por ter muitas requisições IDENT ao mesmo tempo. Padrão de 10 segundos.

ident_timeout segundos

shutdown_lifetime

Quando o Squid recebe um SIGTERM ou um SIGHUP, o cache é colocado em modo de "shutdown pendente" até que todos os sockets ativos sejam fechados. Qualquer cliente ainda ativo depois desse período irá receber uma mensagem de timeout. Default de 30 segundos.

shutdown_lifetime segundos

Tags da seção Access Control Lists

Sem dúvida a parte mais importante para os administradores. Com o uso de ACLs bem configuradas e planejadas, é possível não só manter seus usuários sob controle, mas também melhorar desempenho e facilitar a administração.

acl

Define uma lista de acesso. Quando usando um arquivo para buscar os dados, o mesmo deve conter uma informação por linha. Expressões regulares são case−sensitive − para faze−las case−insensitive, utilize a opção −i.

acl nome tipo string1 ... | "arquivo"

src	Baseado em ip ou hostname de origem da requisição

acl nome src ip/máscara.

dst	Baseado em ip ou hostname de destino da requisição. A ACL só é interpretada depois que a resolução DNS for feita.

acl nome dst ip/máscara.

srcdomain	O domínio da máquina cliente. Os domínios serão obtidos por resolução reversa de IP, o que pode causar atrasos para a resposta da requisição.

acl aclname srcdomain nome_domínio

dstdomain	Mesmo que srcdomain, mas levando−se em conta o destino.

acl nome dstdomain nome_domínio

srcdom_regex	Expressão regular que é avaliada para tentar marcar um domínio requisitante.

acl nome srcdom_regex regex

dstdom_regex	Mesmo que srcdom_regex, mas com relação ao destino.

acl nome dstdom_regex regex

time	Dia da semana e hora

acl nome time [abreviação−do−dia] [h1:m1−h2:m2]

Onde:

- S − Sunday (Domingo)
- M − Monday (Segunda−Feira)
- T − Tuesday (Terça−Feira)
- W − Wednesday (Quarta−Feira)
- H − Thursday (Quinta−Feira)
- F − Friday (Sexta−Feira)
- A − Saturday (Sábado)

h1:m1 − horário de início h2:m2 − horário do término

url_regex	Essa ACL irá procura em na URL uma expressão regular que especificada. Opção case−sensitive

acl nome url_regex regex

urpath_regex	Essa acl irá fazer uma combinação de uma expressão regular com o caminho em um servidor que está se tentando acessar. Isso significa que o Squid irá ignorar o nome do servidor e o protocolo utilizado.

acl nome urlpath_regex regex

port	O acesso pode ser controlado pela porta do endereço do servidor requisitado.

acl nome port numero−porta

proto	Especifica o protocolo de transferência (http, ftp, etc).

acl nome proto protocolo

method	Especifica o tipo de método da requisição.

acl nome method tipo−método

browser	Expressão regular cujo padrão tentara combinar com o contido no cabeçalho HTTP de requisição do cliente, descobrindo assim o agente (browser) utilizado.

acl nome browser tipo

ident	Seqüência de caracteres que combinam com o nome do usuário. Requer um servidor Ident rodando na máquina do cliente.

acl nome ident nome_usuário

ident_regex	O mesmo que ident, mas utilizando−se de expressão regular.

acl aclname ident_regex pattern

src_as	Origem de um sistema autônomo

dst_as	Destino de um sistema autônomo

snmp_community	Comunidade SNMP.

acl snmppublic snmp_community public

maxconn	Limite máximo de conexões provenientes de um mesmo cliente. Útil para restringir número de usuários por IP, bem como fazer controle de uso da banda.

req_mime_type	Expressão regular que combina com o tipo de conteúdo contido no cabeçalho de requisição.

acl nome req_mime_type padrão

arp	MAC Address do cliente.

acl nome arp MAC_ADDRESS

http_access

Permite ou nega acesso ao serviço http baseado na lista de acesso (acl) definida. O uso de "!" indica que será a negação da acl.

Se nenhuma das acls configuradas se encaixar na requisição em curso, será então aplicada a última regra. É importante sempre criar uma acl chamada all (ou descomentar a linha já existente) e colocar um `http_access deny all'.

http_access allow|deny [!]nome ...

icp_access

Permite ou nega acesso à porta ICP, baseando−se nas listas de acesso.

icp_access allow|deny [!]nome ...

miss_access

Usado para forçar seus vizinhos a usar seu servidor como "irmão" ao invés de "pai".

miss_access allow|deny [!]nome...

cache_peer_access

Similar ao `cache_peer_domain', mas oferece mais recursos por utilizar−se da flexibilidade das acls. Sua sintaxe é idêntica ao `http_access'.

cache_peer_access cache−host allow|deny [!]nome ...

ident_lookup_access

Uma lista de elementos em uma ACL, os quais, se encontrados, irão gerar uma requisição IDENT.

ident_lookup_access allow|deny nome ...

Tags da seção auth_param

Uma das principais mudanças do Squid 2.4.x para o 2.5.x foi o sistema de autenticação. Todas as opções referentes a isso estão agoras sujeitas a opção auth_param. Vamos ver abaixo como ela funciona.

Formato geral

auth_param esquema parâmetro [opções]

program

Especifica o programa utilizado para autenticação. Tal programa irá ler uma linha contendo "usuário senha" e responder ao squid com um "OK" para sucesso ou um "ERR" para falha. Para utilizar um autenticador, é necessário uma acl do tipo proxy_auth. Por padrão, utiliza−se o sistema de autenticação básico.

auth_param basic program /path/do/programa /path/do/arquivo/senhas

children

Número de processos filhos que o programa de autenticação poderá conter.

auth_param basic children número

realm

Texto que irá aparecer na caixa de diálogo de login. Não é necessário configurar, mas confere uma certa personalização ao servidor.

auth_param basic realm Texto de login

credentialsttl

Especifica por quanto tempo o Squid irá assumir que uma autenticação bem sucedida continuará válida.

auth_param basic credentialsttl tempo

Tags da seção parâmetros administrativos

O parâmetros configurados nessa seção permitem que o administrador do Squid especifique usuário e grupos no qual o Squid irá rodar, bem como hostname que irá aparecer quando houver erros, etc.

cache_mgr

Usando essa tag, nós podemos especificar o endereço de e−mail do administrador do cache local, que será o responsável pela instalação dessa máquina. Esse usuário será notificado por e−mail caso o cache morra. (usuário local). Padrão: webmaster

cache_mgr usuário

cache_effective_user / cache_effective_group

Quando iniciado como root, o Squid irá procurar esse parâmetro para determinar o usuário e grupo no qual irá rodar. É importante ressaltar que iniciar o Squid com usuário não root fará com que ele não consiga abrir nenhuma porta abaixo de 1024 localmente. Ao configurar esse parâmetro, tenha certeza de que o usuário escolhido terá as permissões necessárias para escrever no diretório de logs, cache e todos os necessários.

cache_effective_user usuário

cache_effective_group grupo

visible_hostname

Se você deseja apresentar uma mensagem de erro com um hostname específico, defina aqui essa opção. Do contrário o Squid irá tentar descobrir o hostname. Esse parâmetro não será necessário se você não tiver um grande cluster de Squids.

visible_hostname nomehost

hostname_aliases

Uma lista de outros nomes que seu cache possa ter. Essa opção é usada para detectar requisições internas quando um cache tem mais de um hostname em uso.

hostname_aliases nomehost

Tags da seção httpd−accelerator

O Squid pode ser usado como um balanceador de carga ou redutor de carga de um webserver em particular. Alguns caches podem trabalhar com requisições de cache e requisições http, fazendo deles também um servidor web. O desenvolvimento do Squid não optou por essa solução. Entretanto, adicionando−se uma camada de tradução o Squid pode receber e interpretar requisições no formato web−server, as quais ele irá repassar ao servidor web real, situado atrás dele.

Nessa seção também configura−se o Squid para trabalhar de modo transparente.

httpd_accel_host

Configura o nome do host para o serviço acelerado. Se você tiver vários servidores, será necessário utilizar a palavra virtual ao invés de hostname.

httpd_accel_host hostname(IP)|virtual

httpd_accel_port

Porta para qual as requisições aceleradas serão enviadas.

httpd_accel_port porta

httpd_accel_single_host

Se você está utilizando o Squid como um acelerador web e tem somente um servidor no backend, configure essa opção para on. Isso fará com que o Squid mande as requisições para o servidor, independentemente do que o cabeçalho disser.

httpd_accel_single_host on|off

httpd_accel_with_proxy

Se a opção http_accel_host estiver ativada, então o Squid irá parar de trabalhar a funcionalidade de cache. É necessário configurar essa opção para que ambas as funcionalidades continuem ativas.

httpd_accel_with_proxy on|off

httpd_accel_uses_host_header

As requisições HTTP/1.1 incluem um cabeçalho relativo ao host, que basicamente contém o nome do mesmo na URL. O Squid pode ser um acelerador para diferentes servidores web através da analise do cabeçalho http. Entretanto, o Squid não checa os valores do cabeçalho do host, abrindo uma possível brecha de segurança. Mais uma vez, é recomendado utilizar essa tag com cuidado.

httpd_accel_uses_host_header on|off

Tags da seção Miscellaneous

Como o nome sugere, essa seção sobre alguns parâmetros que não podem ser explicitamente encaixados com nenhuma outra categoria. Iremos abranger:

- Limite de crescimento de arquivos de log.
- Mostrar informações customizadas sobre os clientes e erros.
- Definir pools de memória para o Squid.
- Gerenciamento por SNMP.
- Coordernação com caches vizinhos através de WCCP.
- Direcionar as requisições tanto para o servidor de origem como um cache vizinho.

dns_test names

O teste de DNS pára de ser executado tão logo ele consegue resolver a primeira busca de nome. Esse teste pode ser desabilitado iniciando−se o Squid com a opção −D na linha de comando.

dns_testnames URL

logfile_rotate

Especifica o número de rotações executadas quando da digitação de `squid −k rotate'. O padrão é 10, o que significa que o Squid criará extensões de 0 até 9. Configurar o log_rotate para 0 irá desabilitar o rotacionamento.

logfile_rotate número

append_domain

Anexa o nome do domínio local para hostnames sem nenhum ponto (.). Essa opção deve conter um domínio com ponto (.) no início.

append_domain nome_domínio

tcp_recv_bufsize

Tamanho máximo de um buffer TCP.

tcp_recv_bufsize bytes

err_html_text

Especifica o texto do HTML que será incluído nas mensagens de erro. Pode ser alguma mensagem sobre contato do administador, ou um link para a página da empresa.

err_html_text texto

deny_info

Essa opção pode ser usada para retornar uma página de erro para requisições que não passem pelas regras definidas em uma ACL. Você pode utilizar as páginas padrão de erro do Squid ou criar as suas próprias.

deny_info nome_pagina_erro acl

memory_pools

Se configurado, o Squid irá manter pools de memória alocada e livre para uso futuro.

memory_pools on|off

memory_pools_limit

Deve−se também determinar um valor para esse pool de memória em bytes. Se não configurada, ou com valor igual a zero, o Squid irá guardar tanta memória quanto possível.

memory_pools_limit bytes

forwarded_for

Atualmente o padrão HTTP/1.1 não provê nenhuma forma de indicar o endereço de requisição de um cliente. Entretanto, como essa era uma feature requisitada, o Squid adiciona em suas requisições um cabeçalho do tipo "X−Forwarded−For". Se ativada essa opção o Squid irá mandar requisições com o IP de origem no cabeçalho. Caso contrário, o mesmo irá ter origem desconhecida.

forwarded_for on|off

log_icp_queries

Configurando−se essa opção como ativa, as requisições ICP passarão a ser logadas no access.log.

log_icp_queries on|off

icp_hit_stale

Se você deseja retornar um ICP_HIT para objetos estáticos cacheados, configure essa opção para `on'.

icp_hit_stale on|off

minimum_direct_hops

Se você utilizar ICMP, faça buscas diretas para sites que estejam a mais de um hop de distância. Esse parâmetro é útil para descobrir a latência da rede.

minimum_direct_hops número

minimum_direct_rtt

Se estiver utilizando ICMP, faça buscas diretas a sites que estejam a mais do que o número de milisegundos configurados aqui de distância. Padrão de 400.

minimum_direct_rtt tempo

cachemgr_passwd

Especifica a senha para operações de gerenciamento de cache.

cachemgr_passwd senha ação ação ...

Ações
5min events non_peers via_headers
60min filedescriptors objects vm_objects
asndb fqdncache pconn
authenticator histograms peer_select
cbdata http_headers redirector
client_list info refresh
comm_incoming io server_list
config ipcache shutdow
counters delay mem store_digest
digest_stats menu storedir
dns netdb utilization
store_avg_object_size (kbytes)

O tamanho médio de objetos é usado para estimar o número de objetos que seu cache pode manipular. Para fazer essa estimativa, basta calcular: Número de objetos = cache_swap/tamanho médio de objetos.

store_avg_object_size tamanho

store_objects_per_bucket

Número de objetos armazenados de uma única vez em uma tabela hash.

store_objects_per_bucket kbytes

client_db

Se você deseja desabilitar estatísticas por cliente, desabilite essa opção.

client_db on|off

netdb_low / netdb_high

Os limites mínimos e máximos da medição ICMP. Por padrão esses valores são 900 e 1000. Isso significa que quando o limite máximo é atingido, o banco de dados irá apagar registros até alcançar o limite mínimo.

netdb_low entradas

netdb_high entradas

netdb_ping_period

O tempo mínimo de medição de um site.

netdb_ping_period time−units

query_icmp

Se você deseja fazer com que as requisições ICP sejam também respondidas com informações ICMP pelos seus vizinhos, habilite essa opção. Lembre−se que é necessário que o Squid tenha sido especificamente compilado com suporte a icmp para que essa opção seja funcional.

query_icmp on|off

test_reachability

Quando habilitado, repostas ICP MISS serão interpretadas como ICP MISS NOFETCH se o host alvo não estiver na base de dados ICMP ou tiver um RTT zero.

test_reachability on|off

reload_into_ims

Habilitando essa opção, você fará com que uma requisição no−cache seja transformada em uma if−modified−since. Essa opção deve ser usada apenas em casos muitos específicos.

reload_into_ims on|off

always_direct

Pode utilizar elementos de uma ACL para especificar requisições que devem sempre ser encaminhadas para o servidor de origem. Isso normalmente é utilizado juntamente com a opção cache_peer.

always_direct allow|deny [!]nome ...

never_direct

É a regra oposta ao always_direct, funcionando da mesma maneira.

never_direct allow|deny [!]aclname ...

anonymize_headers

Substitui o antigo cabeçalho `http_anonymizer' por uma opção mais configurável. Agora é possível especificar quais cabeçalhos serão enviados ou removidos das requisições.

anonymize_headers allow|deny nome_cabeçalho ...

É possível utilizar essa opção permitindo que determinados tipos de cabeçalhos sejam vistos ou negando outros.

Para ter uma header igual ao `http_anonymizer', é preciso configurar da seguinte forma:

- anonymize_headers allow Allow Authorization Cache−Control
- anonymize_headers allow Content−Encoding Content−Length
- anonymize_headers allow Content−Type Date Expires Host
- anonymize_headers allow If−Modified−Since Last−Modified
- anonymize_headers allow Location Pragma Accept
- anonymize_headers allow Accept−Encoding Accept−Language
- anonymize_headers allow Content−Language Mime−Version
- anonymize_headers allow Retry−After Title Connection
- anonymize_headers allow Proxy−Connection

fake_user_agent

Essa opção faz com que o Squid envie, como versão do browser, o parâmetro que for configurado.

fake_user_agent String

icon_directory

Especifica o diretório em que os ícones estão armazenados.

icon_directory path_diretório/nome_diretório

error_directory

Caso deseje customizar as mensagens de erro do Squid, basta indicar o diretório onde os htmls serão encontrados e cria−los de acordo com a padronização.

error_directory path_diretório/nome_diretório

minimum_retry_timeout

Especifica o tamanho mínimo de timeout, quando esse tempo é reduzido para compensar a disponibilidade de múltiplos endereços IP.Isso significa que quando uma conexão é iniciada com um host que tem múltiplos endereços IPs, o tempo padrão de timeout é então reduzido dividindo−se esse valor pelo número de endereços.

minimum_retry_timeout segundos

maximum_single_addr_tries

Configura o número máximo de tentativas de conexões em um servidor que tenha somente um endereço.

maximum_single_addr_triesnúmero

snmp_port

O Squi tem a capacidade de fornecer informações sobre status e estatísticas via SNMP. Aqui configuramos a porta onde esse serviço irá escutar. Utilize 0 para desabilitar essa opção. Padrão: 3401.

snmp_port porta

snmp_access

Permite ou nega acesso à porta SNMP, baseando−se em uma acl.

snmp_access allow|deny [!]aclname ...

Tags da seção delaypool

Conceitualmente, as delay pools são limitantes de consumo de banda. Basicamente o que um delay pool faz é criar uma lentidão artificial para os clientes, gerando uma grande economia de banda. Com uma combinação bem feita de delay pools e acls, é possível fazer um grande controle e limitação de banda.

delay_pools

Número total de delay pools que irão ser utilizadas. Isso significa que se você tiver uma delay pool de classe 2 e 4 de classe 3, esse número deverá ser 5.

delay_pools número

delay_class

Define a classe de cada delay pool. Deve haver exatamente uma classe de delay para cada delay pool.

delay_class número(delay−pool number), número (delay class)

delay_access

Determina em qual delay pool uma requisição será encaixada. A primeira a combinar será utilizada, por isso verifique com cuidado suas acls.

delay_access allow|deny nomeacl

delay_parameters

Define os parâmetros para uma delay pool. Cada delay pool tem um número de alocação de tráfego associado.

- delay_parameters pool agregado (delay_class 1)
- delay_parameters pool agregado individual (delay_class 2)
- delay_parameters pool agregado network individual (delay_class 3)
- delay_initial_bucket_level

Determina qual a porcentagem colocada em cada alocação quando o Squid é iniciado.

delay_initial_bucket_levelbytes

incoming_icp_average / incoming_http_average / incoming_dns_average / min_icp_poll_cnt / min_dns_poll_cnt / min_http_poll_cnt

São descritos os algoritmos usados para as tags acima,

TagName número

Padrão:

incoming_icp_average 6

incoming_http_average

4 incoming_dns_average

4 min_icp_poll_cnt 8

min_dns_poll_cnt 8

min_http_poll_cnt 8

max_open_disk_fds

Especifica o número máximo de file descriptors que o Squid pode usar para abrir arquivos. Essa opção é usada para evitar gargalo de I/O e acesso a disco limitando o número de arquivos.

max_open_disk_fds número

offline_mode

Com essa opção ativada, o Squid nunca irá tentar validar objetos cacheados.

offline_mode on|off

uri_whitespace

A ação que será tomada quando uma URI contiver espaços em branco é decidida nessa tag. Padrão é strip.

uri_whitespace opções

Opções	Descrição
strip	Os espaços em branco são removidas da URL, de acordo com o recomendado na RFC2616
deny	A requisição é negada e o cliente recebe uma mensagem de "Requisição Inválida"
allow	A requisição é aceita e os espaços em branco não são alterados.
encode	A requisição é aceita e os espaços são codificados de acordo com a RFC1738
chop	A requisição é cortada e mandada apenas até o espaço em branco

broken_posts

Uma lista de elementos de uma ACL que, se encontrados, irão fazer com que o Squid coloque um par extra de CRFL (Carriage return e Line Feed) em um PUT ou POST. Isso somente é utilizado junto a alguns servidores HTTP problemáticos que exigem essa modificação. Se não souber de nenhum caso específico, ignore essa opção.

broken_posts allow|deny nomeacl

nonhierarchical_direct

Por padrão, o Squid irá enviar qualquer requisição não hierárquica diretamente aos servidores de origem. Se você desabilitar isso, o Squid irá enviar isso para o cache "pai". Na maior parte dos casos, não é uma boa idéia desabilitar essa opção, visto que ela irá gerar uma latência desnecessária, sem necessariamente algum ganho.

nonhierarchical_direct on|off

prefer_direct

O comportamento normal do Squid é tentar utilizar seus "pais" na maior parte das requisições. Uma possível utilidade de habilitar uma busca direta ao invés disso, seria combinando as opções non hierarchical_direct off and prefer_direct on, fazendo basicamente dos "pais" uma rota backup em caso de erro em buscas diretas.

prefer_direct on|off

strip_query_terms

Para habilitar o log de todos parâmetros das requisições, é necessário habilitar essa opção. Caso contrário o Squid apenas dá forward das mesmas sem gerar um log completo.

strip_query_terms on|off

coredump_dir

Em caso de falhas, os sistemas Unix geram sempre um arquivo de core dos programas. O Squid normalmente guarda os arquivos de core gerados por ele no diretório de cache. Com essa opção é possível configurar onde será armazenado esse arquivo.

coredump_dir diretório

redirector_bypass

Quando habilitado, uma requisição não irá através dos redirecionadores se todos eles estiverem ocupados. Se estiver com essa opção estiver desativada e a fila começar a crescer muito, o Squid irá abortar e gerar um erro solicitando que a quantidade de redirecionadores seja aumentada.

redirector_bypass on|off

ignore_unknown_nameservers

O Squid sempre verifica se uma resposta DNS está sendo recebida de um mesmo IP de origem para qual está sendo enviada a requisição. Caso não sejam os mesmos, o Squid irá ignorar a resposta e mandar uma mensagem no log. Recomendo que não desabilite essa opção, visto que é uma proteção a mais contra ataques baseados em DNS.

ignore_unknown_nameserverson|off

digest_generation

Aqui é possível controlar se o servidor irá gerar um resumo e o tipo de seu conteúdo. Para habilitar essa e todas as outras opções referentes a resumo, é necessário que o Squid tenha sido compilado com opção −−enable−cache−digests.

digest_generation on|off

digest_bits_per_entry

Número de bits do resumo de cache do servidor, o qual será associado com a combinação de um dado tipo de método HTTP e URL.

digest_bits_per_entry número

digest_rebuild_period

Número de segundos para a reconstrução do resumo do cache. O padrão é de 1 hora.

digest_rebuild_period tempo

digest_rewrite_period

Tempo de espera entre escritas de resumo no disco. Como na opção anterior, o resumo é escrito a cada 1 hora.

digest_rewrite_period tempo

digest_swapout_chunk_size

Número de bytes do resumo a escrever de cada vez. Por padrão o Squid utiliza 4KB, que é o tamanho padrão de uma página de swap.

digest_swapout_chunk_size bytes

digesvt_rebuild_chunk_percentage

Configura−se aqui a porcentagem do resumo de cache que será verificada de cada vez. Por padrão está configurado para 10% do total.

digest_rebuild_chunk_percentage porcentagem

chroot

Devido a alguns procedimentos que necessitam de poderes de root, o Squid roda parcialmente como tal. Se você deseja rodar o Squid como chroot, é preciso habilitar essa opção. Isso fará com que o Squid rode os procedimentos necessários como root e depois abandone completamente esse privilégio. Lembre−se que para usar um chroot é necessário um chroot_dir.

chroot enable|disable

client_persistent_connections / server_persistent_connections

Suporte a conexões persistentes para clientes e servidores. Por padrão, o Squid irá usar conexões persistentes para comunicar−se com clientes e servidores.

client_persistent_connectionson|off

server_persistent_connectionson|off

pipeline_prefetch

Para melhorar o desempenho de requisições e fila, o Squid irá trabalhar com 2 requisições paralelamente.

pipeline_prefetch on|off

extension_methods

O Squid somente trabalha com requisições HTTP padrão. Apesar de métodos diferentes serem negados, é possível fazer com que eles sejam aceitos adicionando−os a uma lista. É possível incluir até 20 métodos diferentes.

extension_methods request método

high_response_time_warning

Se a média de falhas por minuto excede esse valor, o Squid manda um aviso de nível 0 no debug (normalmente gerando uma saída no syslog) de alerta.

high_response_time_warningmsec

high_page_fault_warning

Se a média de falhas por minuto excede esse valor, o Squid manda um aviso de nível 0 no debug (normalmente gerando uma saída no syslog) de alerta.

high_page_fault_warning time−units

high_memory_warning

Se o uso de memória excede o valor determinado, o Squid manda um aviso de nível 0 no debug (normalmente gerando uma saída no syslog) de alerta.

high_memory_warning número

store_dir_select_algorithm

O Squid pode trabalhar com 2 tipos de algoritmos para escolher entre vários diretórios de cache: least−load e round−robin. O padrão é leat_load.

store_dir_select_algorithm tipo_algoritmo

ie_refresh

O Microsoft Internet Explorer até a versão 5.5SP1 tem problemas ao trabalhar com proxy transparente, impossibilitando forçar um refresh. Ativando essa opção é possível corrigir parcialmente o problema, fazendo com que todos os pedidos de refresh vindo de um IE seja automaticamente interpretado como forçado. A melhor opção, quando possível, é atualizar os clientes.

ie_refresh on|off

Outras referências e leituras complementares

http://squid.visolve.com/squid24s1/contents.htm

http://br.groups.yahoo.com/group/squid−br/

http://lasdpc.icmc.sc.usp.br/pesquisa/jac/quali.pdf

http://directory.google.com/Top/Computers/Software/Internet/Servers/Proxy/Caching/Squid/?tc=1/

http://hermes.wwwcache.ja.net/servers/squids.html

http://freshmeat.net/search/?q=squid

http://www.pop−pb.rnp.br/proxy/pal0100.PPT

http://squid.visolve.com/squid24s1/externals.htm#authenticate_program

http://web.onda.com.br/orso/ncsaplus.html

http://www.hacom.nl/%7Erichard/software/smb_auth.html

http://www.tldp.org/HOWTO/Bandwidth−Limiting−HOWTO/index.html

http://squid.visolve.com/squid24s1/access_controls.htm

http://stein.cshl.org/WWW/software/GD/

http://squid.visolve.com/squid24s1/contents.htm

http://br.groups.yahoo.com/group/squid−br/

http://lasdpc.icmc.sc.usp.br/pesquisa/jac/quali.pdf

http://directory.google.com/Top/Computers/Software/Internet/Servers/Proxy/Caching/Squid/?tc=1/

http://hermes.wwwcache.ja.net/servers/squids.html

http://freshmeat.net/search/?q=squid

http://www.pop−pb.rnp.br/proxy/pal0100.PPT

This HTML page is (see source)